BUY IN PRINT WORLDWIDE MAGPI.CC/STORE

The MagPi

Issue 78 | February 2019 | magpi.cc

The official Raspberry Pi magazine

ARCADE GAMES

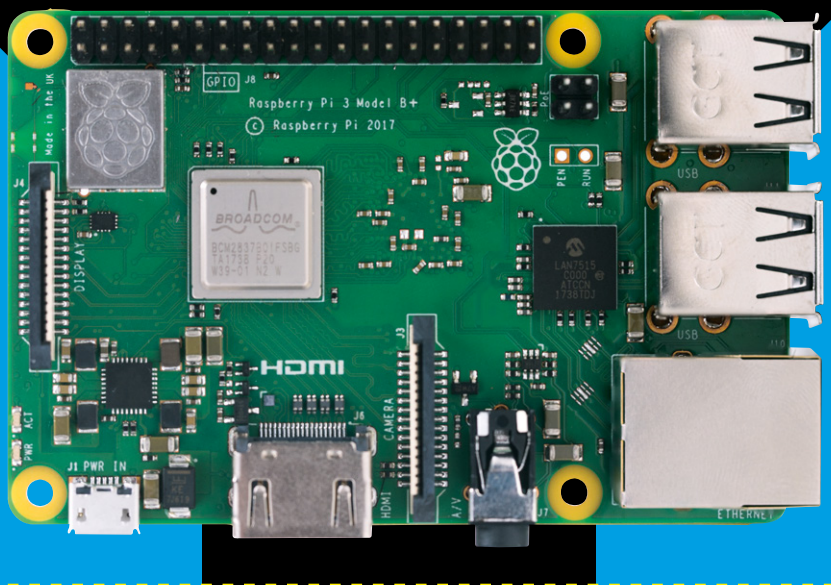
BUILD
A Pinball Machine

CODE
A 3D Game

PLAY
PC Games

Plus!

- Hack a vintage television
- Build a telephone exchange
- Make an MP3 player



Top 10 wearables

Wear your Pi with pride

Social APIs

Plug your Pi into social media

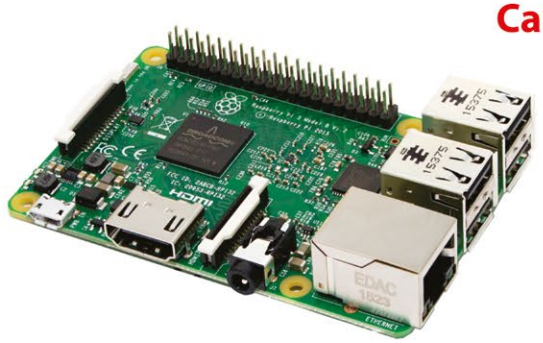
£5.99

GLOBAL DELIVERY
magpi.cc/store

Raspberry Pi PRESS

WIN! PiDP-11 Retro Computers

CanaKit™



CanaKit Raspberry Pi 3 Ultimate Starter Kit

Model B | 1 GB RAM | 1.2 GHz | Quad-Core CPU

> Learn to Code
> Explore Computing
> Get started with Electronics

KIT INCLUDES RASPBERRY PI 3 AND ...

- PREMIUM CASE & HEAT SINKS
- 2.5A POWER ADAPTER
- 32 GB CLASS 10 MICROSD CARD — PRE-LOADED WITH OPERATING SYSTEM
- USB MICROSD CARD READER
- PREMIUM HDMI CABLE
- QUICK-START GUIDE
- GPIO TO BREADBOARD INTERFACE BOARD
- RIBBON CABLE
- FULL-SIZE BREADBOARD
- JUMPERS — MALE TO MALE & MALE TO FEMALE
- LEDS
- RESISTORS & PUSH-BUTTONS

Available for worldwide shipping at:

WWW.CANAKIT.COM

Raspberry Pi Zero W

Now available at CanaKit!

Wi Fi™ Bluetooth®

ELECTRONIC KITS • ELECTRONIC PARTS • RASPBERRY PI • ARDUINO

WELCOME
to The MagPi 78

Most of *The MagPi* team were raised with arcade games of one kind or another. So when we heard about Martin Kauss's Pi-powered pinball machine (page 30), we had to know more.

A lucky few remember the golden age of Williams vs. Bally/Midway, but even if you've never played pinball, there's a lot to learn from building a ball basher.

Younger fans will be more stunned by Steam Link (page 38). With this software installed, you can stream the very latest PC games on a Raspberry Pi.

With Steam Link, a Raspberry Pi computer becomes an amazing alternative to the latest consoles. It's especially incredible when you remember these games are running on a machine costing just $35.

Fun though all of this is, it's far more rewarding to make your own games. We've been looking at coding classics such as Space Invaders, Pong, and Pac-Man recently. Now it's time to build your own game from the ground up: Mark Vanstone starts his tutorials on coding a 3D isometric adventure game (page 42).

We learn by building. And whatever your favourite flavour of arcade game, you'll find something special to build this month.

Lucy Hattersley Editor

EDITOR

Lucy Hattersley

Editor of *The MagPi*. Lucy codes, crafts, and creates wonky robots. She speaks French (badly) and mangles the piano. One day she'll get that pet dog.

magpi.cc

GET A NEW RASPBERRY PI 3A+ WITH A SUBSCRIPTION! PAGE 26

Contents

Issue 78 > February 2019

Cover Feature

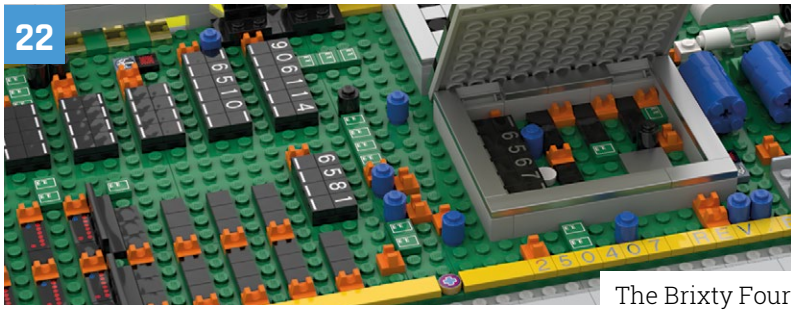
28 Arcade games

Regulars

06 The world of Pi
92 Your letters
97 Next month
98 Final word

Project Showcases

14 The computer mouse
18 Vintage television
20 R3-14 personal assistant
22 The Brixty Four
24 Raspberry Puitar

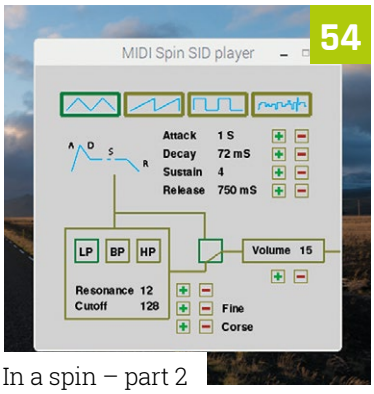


The Brixty Four

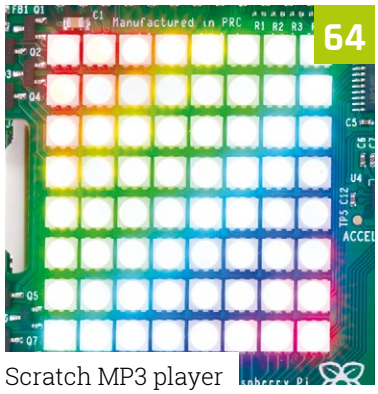
Vintage television

Raspberry Puitar

DISCLAIMER: Some of the tools and techniques shown in The MagPi magazine are dangerous unless used with skill, experience, and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. Children should be supervised. Raspberry Pi (Trading) Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in The MagPi magazine. Laws and regulations covering many of the topics in The MagPi magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in The MagPi magazine may go beyond. It is your responsibility to understand the manufacturer's limits.



In a spin – part 2



Scratch MP3 player

Inky wHAT

Lisa Rode interview

Tutorials

50 Build a telephone exchange
54 In a spin – part 2
60 Easy compilation
64 Scratch MP3 player

The Big Feature

69 Social media APIs

Reviews

74 TinyPi Pro
76 HiFiBerry DAC+ ADC
79 Inky wHAT
80 Java resources
82 Top 10 wearables

Community

84 Lisa Rode interview
86 This month in Raspberry Pi
90 Events

WIN ONE OF TWO **PiDP-11** Retro computer kits

Compute Module 3+ launched

Raspberry Pi lifts the lid on its latest industrial computer. By **Lucy Hattersley**

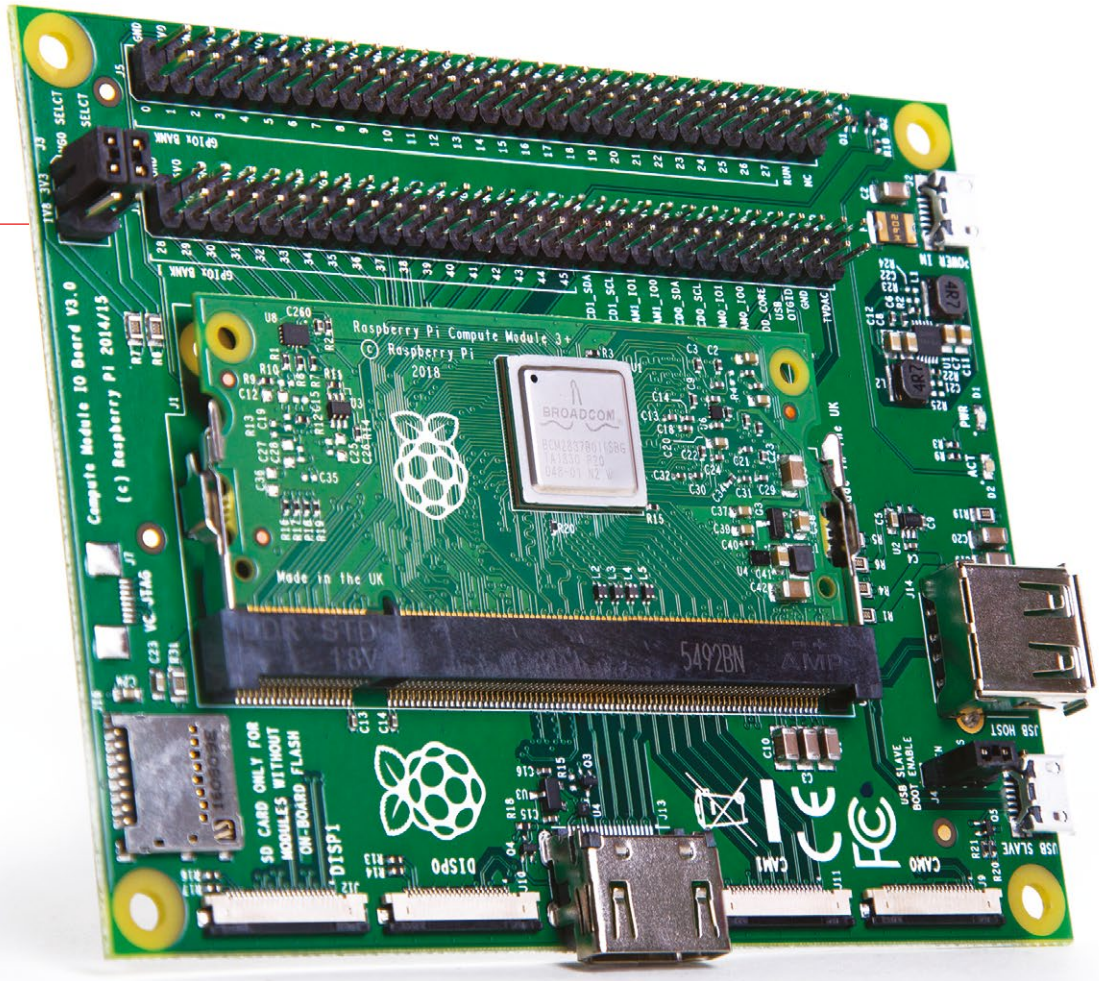
Raspberry Pi has launched a Compute Module 3+ board featuring technical enhancements inspired by recent boards.

Aimed at businesses, and industrial users, the Compute Module is used to embed a Raspberry Pi into systems and commercial products. The Compute Module uses a standard DDR2 SODIMM (small outline dual in-line memory module) form factor. GPIO and other I/O functions are routed through the 200 pins on the board.

The new Compute Module 3+ sports the Broadcom BCM2837B0 chip, albeit clocked at 1.2GHz and with 1GB LPDDR2 SDRAM.

"As always, we try to learn lessons from the market," says Eben Upton, CEO and Raspberry Pi co-founder.

"We haven't increased the rated clock speed, as this is dependent on having the MaxLinear PMIC, which won't fit into the Compute Module 3+ form factor," explains Eben.



▼ The Compute Module I/O board helps with development by providing the various ports you would find on a Raspberry Pi

Compute Module 3+ specs

Processor:
Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.2GHz

Memory:
1GB LPDDR2 SDRAM

Flash memory (eMMC):
8GB / 16GB / 32GB

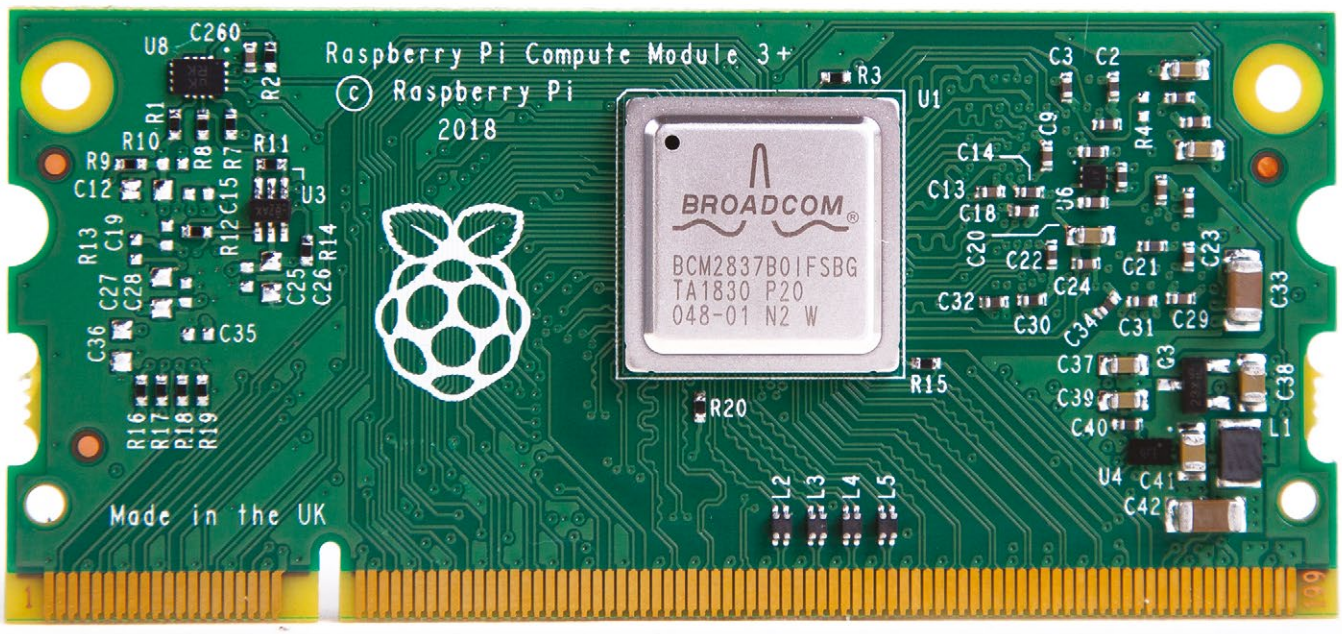
Multimedia:
H.264, MPEG-4 decode (1080p30), H.264 encode (1080p30), OpenGL ES 1.1, 2.0 graphics

Operating temperature:
-20 to +70˚C

Pricing:
$30 (8GB), $35 (16GB), $40 (32GB), $25 (CM3+/Lite)

Net weight:
9g (board only)

Dimensions:
Board; 67.6 × 31.1 × 3.7mm



Rather than speed things up, the aim is to run cooler: "Compute Module 3+ brings improved thermal performance and some tweaks to the PCB (printed circuit board) from Raspberry Pi 3B+."

Cool running

Compute Module 3+ is an "opportunity to deploy designs in environments with a wider range of temperatures," says Eben, and engineers can "drive the chip harder, for longer, without hitting thermal limits." The operating temperature range now ranges from -20 to +70 degrees C.

> Drive the chip harder, for longer, without hitting thermal limits

The new models also boast expanded on-board storage. Models are available with 8GB, 16GB, and 32GB of on-board eMMC flash memory.

A Compute Module 3+ Lite option remains available. Users wire the module pins up to an eMMC or SD card. "Lite accounts for around a quarter of Compute Module sales," reveals Eben.

Customers have been requesting larger storage options, and the new models are sure to be popular with media and data-logging developers. "With flash prices falling, it felt like a good time to address this," says Eben. "We'll have Lite, 8GB, 16GB, and 32GB options at $25, $30, $35, and $40 respectively. At the high end, that's a lot of fun for $40."

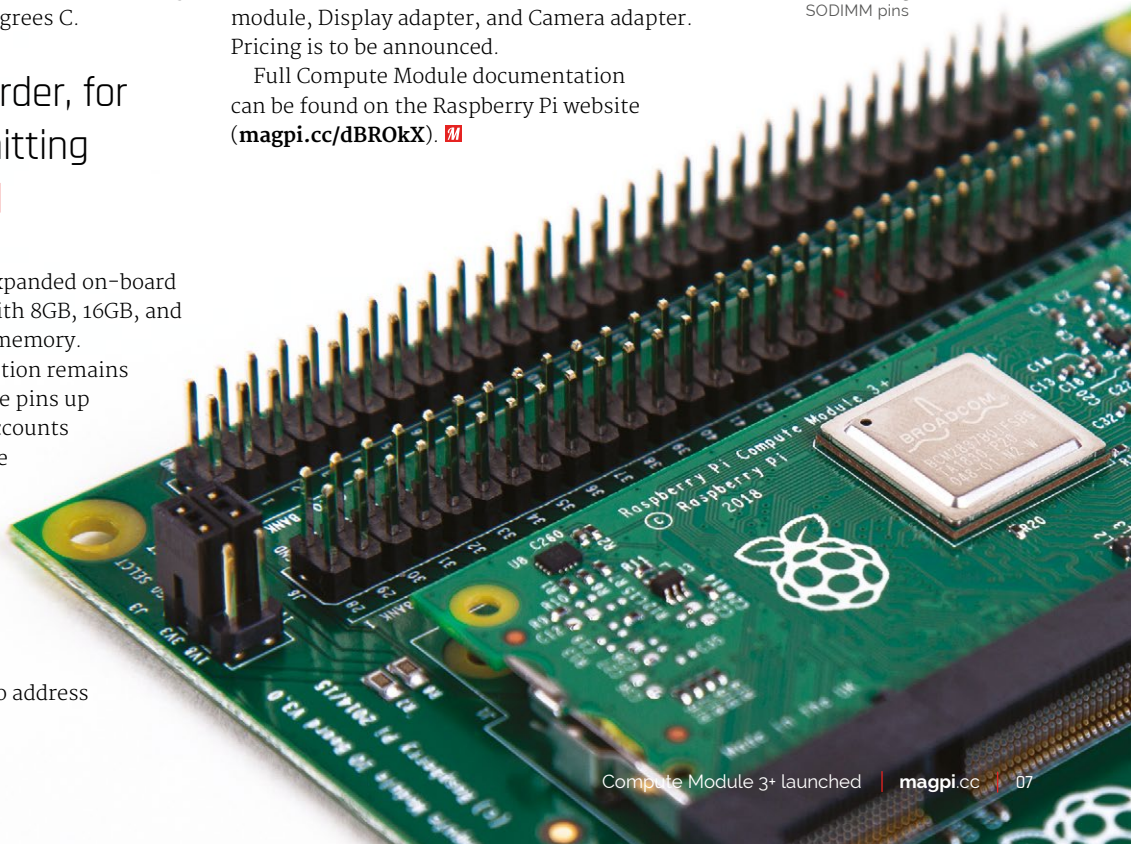
A separate Raspberry Pi Compute Module I/O board can be purchased to aid development. This breakout board provides the various ports you would find on a Raspberry Pi (such as GPIO pins, HDMI, USB, Camera, and Display).

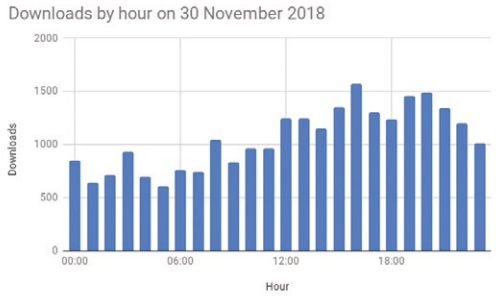
A Raspberry Pi CM3+ Development kit will also be available. It consists of a Raspberry Pi Compute Module I/O board, CM3+/32GB module, CM3+/Lite module, Display adapter, and Camera adapter. Pricing is to be announced.

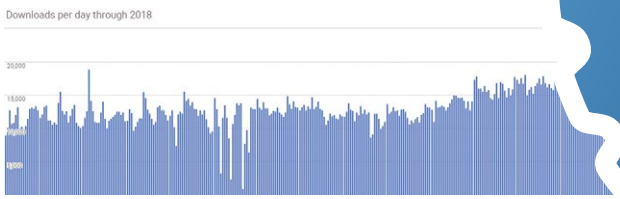
Full Compute Module documentation can be found on the Raspberry Pi website (**magpi.cc/dBROkX**).

▲ The Broadcom BCM2837B0 chip and board refresh enable to the board to operate in a wider temperature range

▼ The Compute Module uses a standard DDR2 SODIMM form factor with GPIO and other functions routed through the 200 SODIMM pins







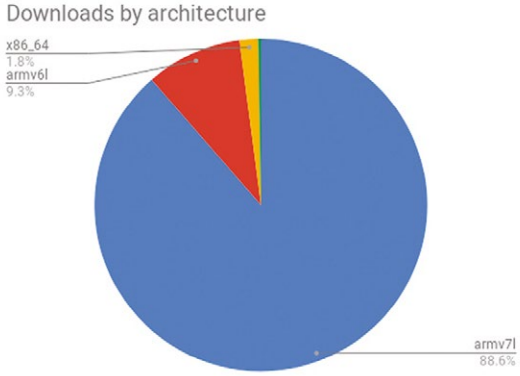
Saving lives with piwheels

▲ The piwheels project has made it much easier for Raspberry Pi users to download projects

A year's worth of data on which Python package downloads, and how much build time, the piwheels project has saved users. By **Ben Nuttall**

The piwheels project is a Python package repository providing natively compiled wheels for the Raspberry Pi's ARM architecture.

It makes pip installations much faster for Raspberry Pi users. In 2018, a total of 5,154,233 packages were downloaded, saving users 39 years, 10 months, and 29 days!



Top 30 downloads

The top five most downloaded packages of the year are pycparser, home-assistant-frontend, PyYAML, cffi, and MarkupSafe. Other notable packages in the top 30 include numpy, cryptography, RPi.GPIO, matplotlib, tensorflow, and opencv.

The piwheels project usage was 89% ARMv7 (Pi 2 and Pi 3), 9% ARMv6 (Pi 1 and Pi Zero), and 2% other architectures. By far, most downloads come from Raspbian, particularly Raspbian Stretch. The next highest are Debian Stretch, and then various Ubuntu distributions.

> In 2018, a total of 5,154,233 packages were downloaded

Usage is rising

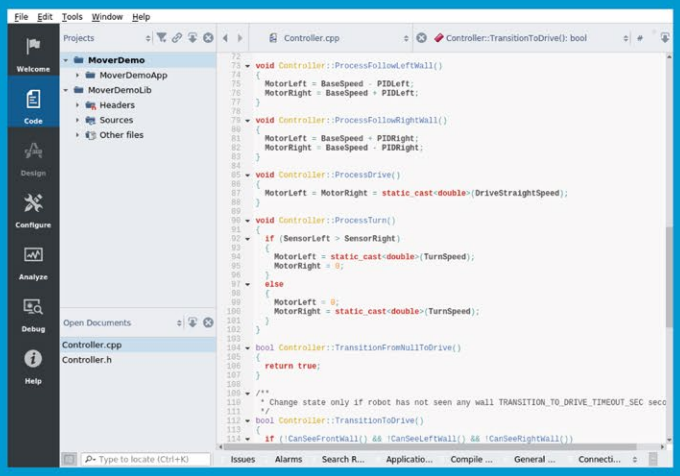
In December, 662,621 packages were downloaded, saving over six years of build time in just one month.

3 ISSUES FROM £5

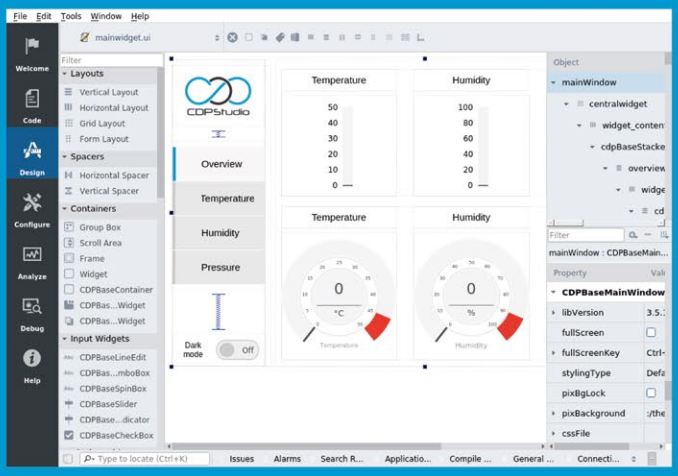
on a quarterly subscription

Use the code MP-SAVE at checkout

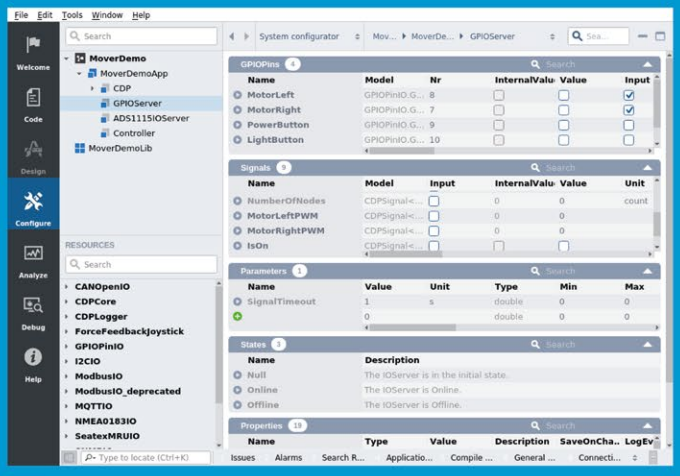
Visit: magpi.cc/345



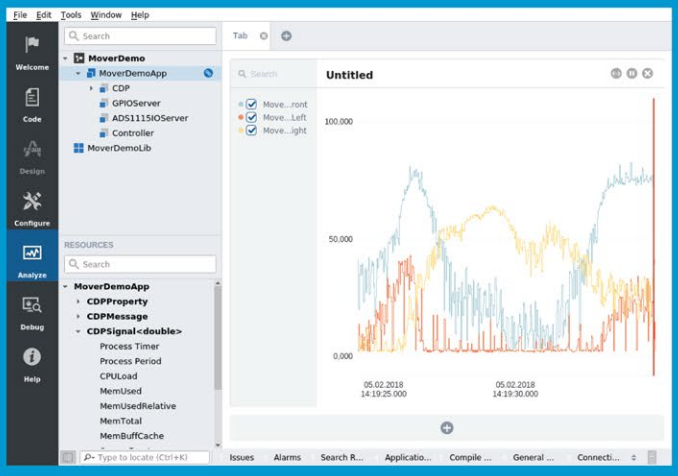
Code



Design



Configure



Analyze

Now free for home projects

A professional control system development tool

CDP Studio is a development platform for industrial control systems, now coming with a free version for non-commercial use. The system can run on a Raspberry Pi, supports C++, open source libraries and has a large feature toolbox including GPIO, I2C and MQTT. Its built in GUI design tool and features lets you code less and do more.

Free download on www.cdpstudio.com

CDP Technologies AS
Nedre Strandgate 29
P.O. Box 144
NO-6001 Ålesund, Norway

Tel: +47 990 80 900
info@cdptech.com
www.cdpstudio.com

CDPStudio

Pi Wars 2019 tickets on sale

Competitors are set to engage in a space race. **Rosie Hattersley** reveals the new robot challenges

Cambridge Computer Laboratory (CCL) is gearing up to host a weekend of space-themed action when it plays host to Pi Wars 2019. Visitors on the last weekend of March will witness all manner of furious Raspberry Pi robot challenges in CCL's William Gates Building.

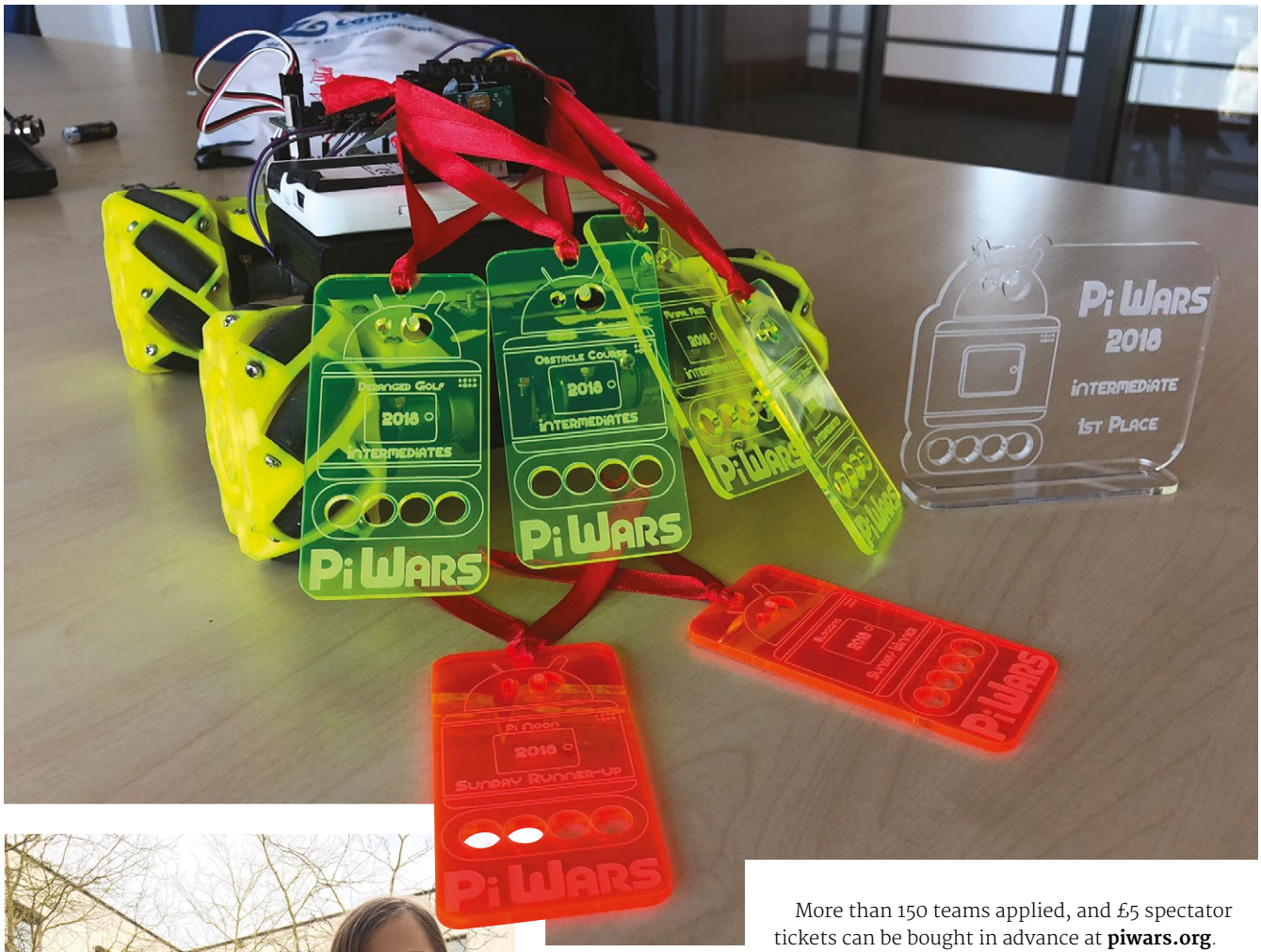
For the uninitiated, Pi Wars is an international, challenge-based robotics competition. Teams build Raspberry Pi-controlled robots and compete in various non-destructive challenges to earn points.

Thanks to the creativity of teams around the world, the Pi Wars stage is packed with a wild array of moving creations. Manoeuvring Pi-mobiles round obstacle courses will be trickier than ever due to new obstacles and brand-new tracks, all designed by the Pi Wars organisers Tim Richardson and Michael Horne.

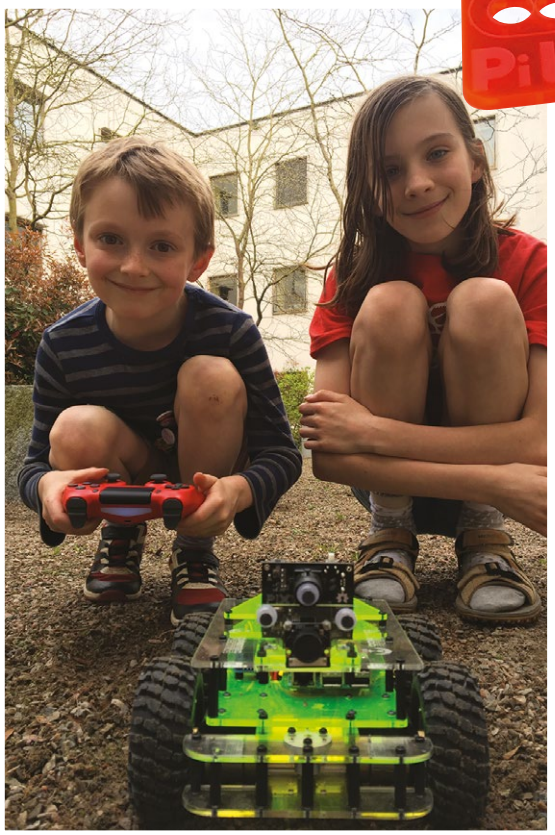
Tim tells us, "I'm particularly looking forward how the robots tackle the Spirit of Curiosity course – a completely new, fetch-and-carry course for this year which has been the hardest build so far."

There's also a new autonomous maze course, an obstacle course, and a kink in the straight-line speed test, keeping each challenge fresh for previous entrants.

▼ Roboteers must navigate a crew back to Earth on the Apollo 13 obstacle course



▲ Prizes are available for the best robots in each group



◀ Adam and Natalie test out Brian Corteil's latest Coretec robot

More than 150 teams applied, and £5 spectator tickets can be bought in advance at **piwars.org**. Tickets on the day are £7 apiece. If you're a volunteer or under 18 years old, you get to enjoy Pi Wars for free.

> Pi-controlled robots compete in various non-destructive challenges

To become a volunteer, sign up here: **magpi.cc/peHmDe**. You'll only be expected to help for two to three hours, after which you'll have plenty of free time to enjoy the rest of the event.

Tim and Mike Horne – and *The MagPi* crew – are very excited to see how this year's Space Exploration theme will be interpreted by Pi Wars entrants.

Tim tells us that more than one six-wheeled Curiosity-evoking vehicle will be make its debut. Pi-based challengers will be judged by ex-*Robot Wars* judge Dr Lucy Rogers, as well as a yet-to-be-announced additional celebrity adjudicator.

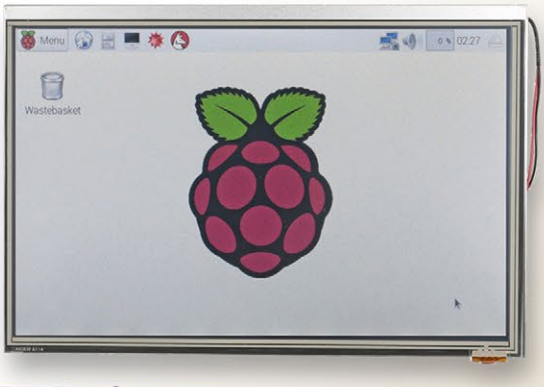
MIDAS

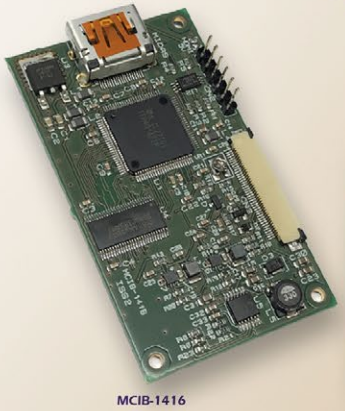
Displays

design • manufacture • supply

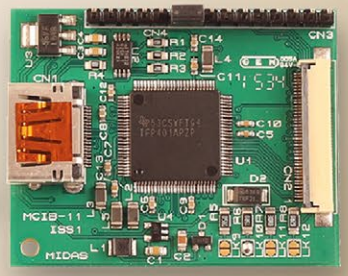
HDMI TFTs

Midas offer a range of HDMI TFT Displays which can be used as 'plug and play' with a Raspberry Pi SBC. Midas offer 5", 7" and 10.1" size options, with each size range coming in either non-touch, capacitive touch, or resistive touch.





MCIB-1416



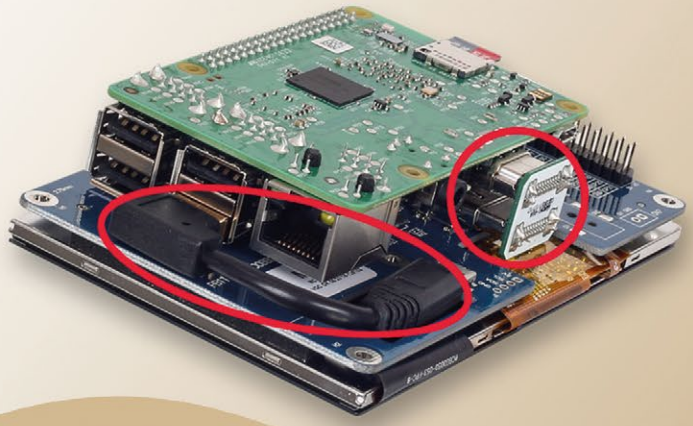
MCIB-11

HDMI TFT Key features:

- 5", 7" and 10.1" size options
- Capacitive touchscreen, resistive touchscreen and non-touchscreen options
- 'Plug and play' with Raspberry Pi
- HDMI interface
- -20 to + 70 wide temperature range
- HDMI connector
- Micro B USB connector for touch panel

Interconnect Solutions for HDMI TFT range

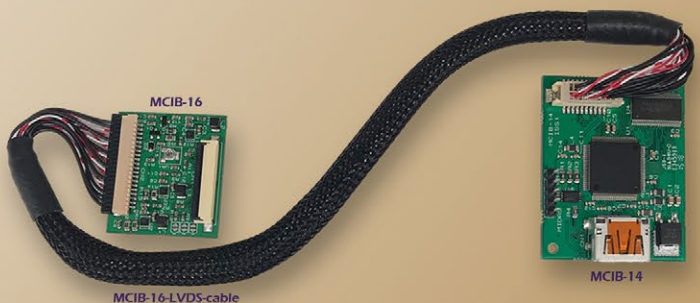
Midas offer two interconnect solutions to complement the range of 'plug and play' HDMI TFTs. The **HDMI to HDMI** interconnect board enables HDMI output from the Pi to the display, and the **USB cable** allows the touchscreen function to be used.



Interconnect Solutions for other TFT displays

Midas also offer a range of boards that provide conversion between HDMI and RGB/LVDS.

- **MDIB-11** This board provides conversion from HDMI to RGB with the addition of EDID (Extended display identification data) and an LED backlight driver.
- **MCIB-1416** This board combines the features of the MCIB-14 and MCIB-16 eliminating the need for an LVDS cable.
- **MCIB-14** This board provides conversion from HDMI to LVDS with the addition of EDID.
- **MCIB-16** This board can be used in conjunction with the MCIB-14 (using MCIB-16-LVDS-cable) to provide pin compatibility and the various voltages (AVDD, VGH, VGL, VCOM and LED BL) required by a range of TFTs from Midas.



MCIB-16-LVDS-cable

All products available from:

Farnell element14

www.farnell.com

midasdisplays.com
+44 (0) 1493 602602
sales@midasdisplays.com

MAKER

Thomas (Electronic Grenade)

A high school student who has been building things his whole life. He got into Raspberry Pi about four years ago, and has since become a tech YouTuber.

magpi.cc/egAQpE

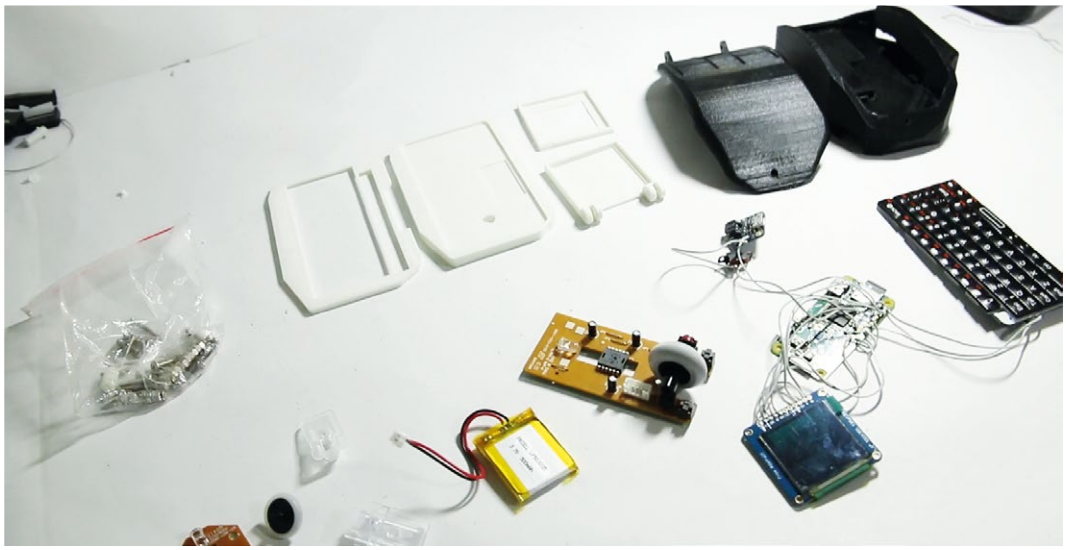
The computer mouse

Look up in the sky! Is it a mouse? Is it a screen? No, it's computer mouse! **Rob Zwetsloot** gets the scoop on our newest project hero

When the Raspberry Pi Zero first launched, we all thought about stuff we could make with it. Some of those can be found way back in issue 40's Pi Zero launch feature (**magpi.cc/40**). However, one we didn't talk about was Eben Upton's idea: the Raspberry Pi co-creator wondered if there was a way to play Doom with a plug-and-play mouse.

While perhaps a little tricky with no keyboard, it's an interesting idea, and we've seen versions of it since then. This project by Thomas (aka Electronic Grenade) takes the concept to a whole new level, though, by putting an entire computer into a mouse.

A tiny fold-out screen that shows you what your mouse is doing
For the full computer experience, a keyboard is included
Don't worry, it has the full functions of a normal mouse – such as scrolling



The many parts needed to build a computer mouse

"My latest project is called 'the computer mouse'," Thomas tells us. "It is a fully-fledged computer in the shape of a mouse. It has a keyboard, a screen, and works like a normal mouse. This project uses the Raspberry Pi Zero W along with an OLED display and a Bluetooth keyboard."

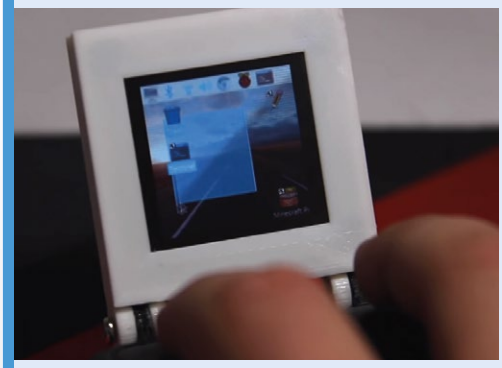
With the ease of setting up Bluetooth devices, and the ability to connect to a wireless network without any dongles, the Pi Zero W was the natural choice for Thomas.

Always test the parts in your project before final construction

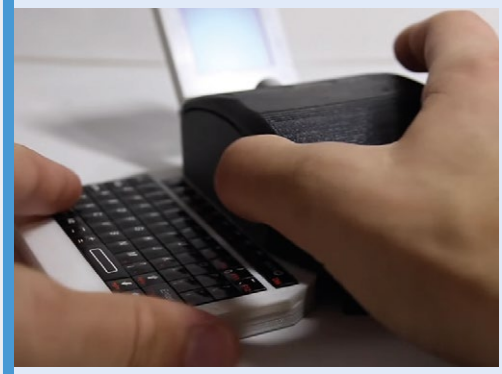
Quick FACTS

- The computer mouse project took about two months to complete
- The 3D-printed parts took an entire month to design
- The screen is only 1.5-inch, and can fold down for easy transport
- Two mice were used for parts in the build
- Thomas has also built a Raspberry Pi-powered laptop

Using a computer mouse



01 Flip out the screen into a position you can see properly. You may have to get a little closer to the mouse to make out everything on the screen, though.



02 Get ready to use your computer mouse by sliding out the keyboard. It's a little wobbly, so make sure to stabilise the entire apparatus by holding the mouse steady.

03 Switch on the computer mouse. You'll have a to be a little patient as it boots up, and then you're ready to use it. Thomas doesn't recommend using Minecraft Pi, though, as it tends to crash after 15 seconds.

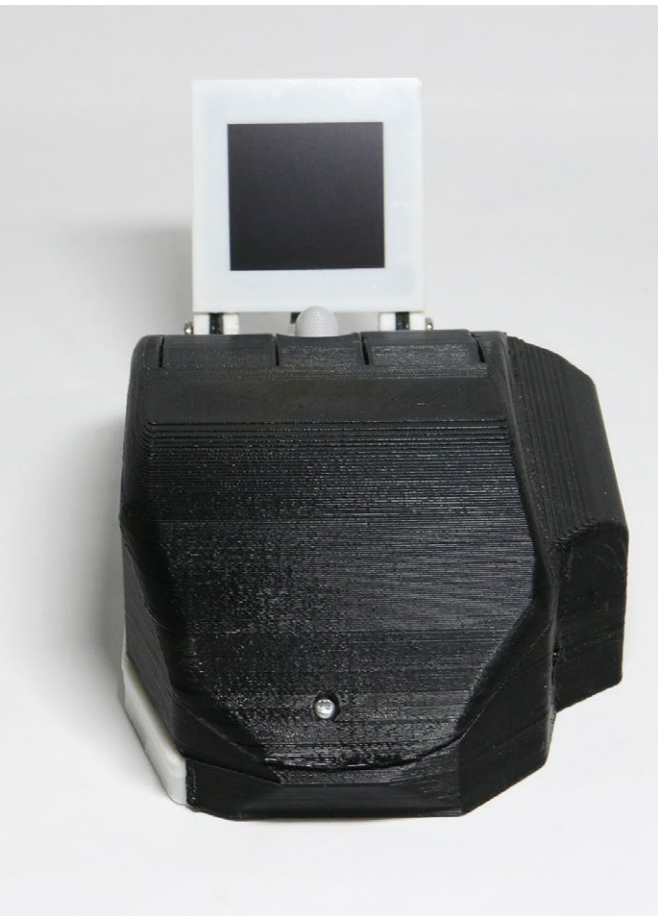
"I thought of this idea after seeing a project where somebody put a Raspberry Pi and a battery into a keyboard," he explains. "It had an HDMI cable coming out of the keyboard that plugged into a monitor so that it could be used as a fully-fledged computer. After seeing that, I thought

> "It takes around 30 seconds to boot up, and you can run most applications just fine"

it would be neat to try and do the same but with a mouse. But instead of having to plug it into a monitor and keyboard, both of those things would be part of it.

"When I started, I envisioned it looking like an ordinary mouse, but with a sliding keyboard and a fold-out monitor. After I got the mouse and all of

▼ The screen is truly tiny, so you may well need to get up close



▲ All the electronics are squeezed into the mouse's chassis

▼ It's not really that much bigger than a standard mouse

the parts, though, I realised that it wasn't going to fit into the mouse. So I started designing my own mouse in Fusion 360, which is a 3D CAD program. After multiple redesigns, I printed it out and put all of the parts inside."

A unique project

The parts of the original mouse were stored inside the new mouse chassis, along with the tiny screen and the pop-out Bluetooth keyboard.

"It works pretty much as expected. It takes around 30 seconds to boot up, and you can run most applications just fine," Thomas says. "There are a couple of design flaws that I would definitely change if I decide to make a second attempt at this. The mouse's sensor isn't very sensitive due to the way I have it in the housing, which makes it sometimes not sense movement. I would definitely make some improvements to that. I would also redesign the tracks that the keyboard slides on to be sturdier. As it is now, the keyboard is very loose feeling when it is extended."

To be honest, we want a kit of this mouse, so we're hoping he works out the kinks. Otherwise, we're looking forward to seeing what other cool and crazy stuff Thomas makes in the future.

Vintage television

Martin Mander's latest project uses the new TV HAT to bring analogue control to digital television. **David Crookes** tunes in

MAKER

Martin Mander

Martin works as an analyst in Norwich, but he loves to breathe new life into vintage technology by combining them with new devices.

magpi.cc/fetqPS

When Martin Mander's portable Hitachi television was manufactured in 1975, there were just three UK channels and you'd need to leave the comfort of your sofa in order to switch between them.

Today, we have multiple viewing options and even a cool Raspberry Pi TV HAT that lets us enjoy DVB-T2 broadcasts via a suitable antenna. So what did nostalgia-nut Martin decide to do when he connected his newly purchased TV HAT to the Pi's 40-pin GPIO header? Why, he stuck it in his old-fashioned TV set with a butt-busting rotary switch and limited the number of channels to those he could count on one hand – dubbing it "the 1982 experience" because he wanted to enjoy Channel 4 which was launched that year.

Going live

Martin is a dab hand at CRT television conversions (he's created six since 2012, using monitors, photo frames, and neon signs to replace the displays). "For my latest project, I wanted to have some fun with the new HAT and see if I'd be able to easily display and control its TV streams on some of my converted televisions," he says. It's now being promoted to his office, for some background viewing as he works. "I had great fun getting the TV HAT streams working with the rotary dial," he adds.

Although Martin jumped straight into the HAT without reading the instructions or connecting an aerial, he eventually followed the guide and found getting it up-and-running to be rather straightforward. He then decided to repurpose his Hitachi Pi project (**magpi.cc/vVobwP**), which he'd already fitted with an 8-inch 4:3 screen.

"It's powered by a Pi 3 and it already had the rotary dial set up and connected to the GPIO," he explains. "This meant I could mess about with the TV HAT, but still fall back on the original project's script if needed, with no hardware changes required."

Change the channel

Indeed, Martin's main task was to ensure he could switch channels using the rotary dial and this, he says, was easier to achieve than he expected. "When you go to watch a show from the Tvheadend web interface, it downloads an M3U playlist file for you which you can then open in VLC or another media player," he says.

> "I had great fun getting the TV HAT streams working with the rotary dial"

"At first, I thought the playlist file was specific to the individual TV programme, as the show's name is embedded in the file, but actually each playlist file is specific to the channel itself so it meant I could download a set of playlists, one per channel, and store them in a folder to give me a full range of watching options."

Sticking to his theme, he stored playlists for the four main channels of 1982 (BBC1, BBC2, ITV, and Channel 4) in a folder and renamed them channel1, channel2, channel3, and channel4.

"Next, I created a script with an infinite loop that would look out for any action on the GPIO pin that was wired to the rotary dial," he continues.

▶ The project was made possible thanks to the new Raspberry Pi TV HAT

The Hitachi television is fitted with a Pimoroni 8-inch 4:3 screen and a Raspberry Pi 3

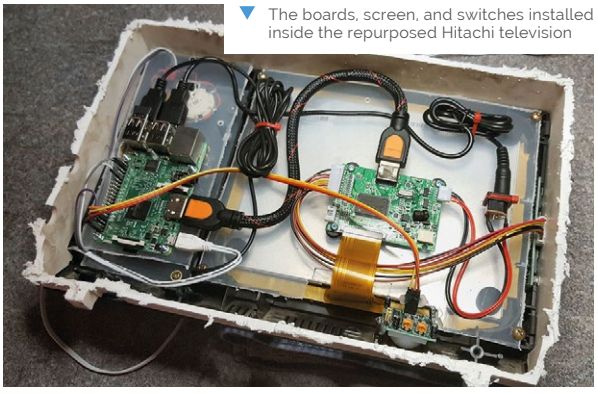
Programmes stream from a Pi 2 server and the channels are changed by turning the dial

The name of the channel briefly appears at the bottom of the screen – the playlist files are edited in Notepad

▲ A young Martin Mander decides the blank screen of his black and white Philips TX with six manual preset buttons is preferable to the shows (but he'd like to convert one of these in the future)

"If the script detects that the switch has been moved, then it opens the first playlist file in VLC, full-screen. The next time the switch moves, the script loops around and adds '1' to the playlist name, so that it will open the next one in the folder."

Martin is now planning the next stage of the project, considering expanding the channel-changing script to include streams from his IP cameras, replacing a rechargeable speaker with a speaker HAT, and looking to make the original volume controls work with the Pi's audio. "It been really satisfying to get this project working and there are many possibilities ahead," he says.



▼ The boards, screen, and switches installed inside the repurposed Hitachi television

Quick FACTS

- An Ethernet-connected Pi 2 acts as a TV server
- Streams are sent to a Pi-infused retro television
- A rotary dial changes channels on the 8-inch display
- A Python script handles channel changing
- Standard and high-definition streams are possible

R3-14 Personal Robot Assistant

As well as a friendly face, this robotic PA offers a host of useful features. **Phil King** discovers how it came to life

MAKER

Sanjeet Chatterjee

A Year 12 student with a passion for technology and design. In his free time, he enjoys programming, tinkering with electronics, and playing the violin.

magpi.cc/BQLMaD

R3-14 stands for 'Raspberry Pi' (geddit?), and this is the debut Pi project of Year 12 student Sanjeet Chatterjee. Seeking to create something original, he settled on a robot personal assistant which would have a physical presence in the room, with interaction being an important feature.

"In the beginning, I did not have a concrete plan," says Sanjeet, "but slowly built up the features, whilst brainstorming different designs and ideas, before finally putting everything together, after having a clear vision of the final product."

He's certainly packed in a lot of features. As well as acting as a smart speaker, using Google Assistant to respond to spoken queries, R3-14 has an impressive range of capabilities. Siri voice commands can be used to turn lights and home appliances on and off via 433MHz transmitters/receivers; a web app

> Robot assistants could one day become the centrepiece in our lives

provides remote control of the robot's motorised head and its on-board camera; while a face-tracking feature enables the head to automatically follow the user while responding with various messages.

Personally engaging

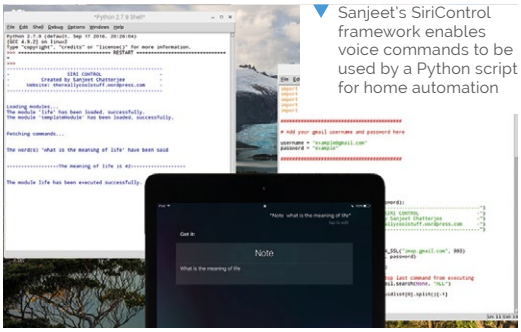
Creating a friendly-looking robot to interact with users was a key factor for Sanjeet, who – during a four-month build process – iterated through various designs, by sketching and creating an aluminium prototype, before settling on the current version, which was built using 3D-printed PLA parts.

"It was important for it to be relatable, with human-like features, whilst also having a distinct design," he explains. "For example, simple features, such as the fading RGB LEDs, which provide user feedback during voice interaction, are seen as 'eyes' by us, even though they are nothing like our eyes."

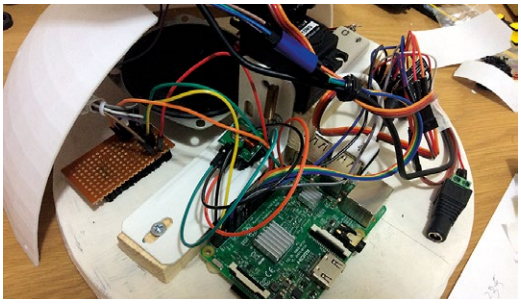
Another level of interaction was the face tracking, which uses OpenCV, with the pre-trained face classifier, for detecting faces in the live video stream from the camera.

Sanjeet's favourite feature, however, is the inclusion of his own SiriControl open-source software framework (**magpi.cc/2t3Bh4v**) – featured in *The MagPi* #60 – which adds another dimension to home automation. "I can ask Siri to control my devices, from anywhere in the world!"

Essentially, SiriControl works by Siri creating a note, which syncs with a Gmail account, before being retrieved through the IMAP protocol with Python. The corresponding module is executed



Sanjeet's SiriControl framework enables voice commands to be used by a Python script for home automation



The base contains a Pi 3, speaker, and a whole lot of wiring

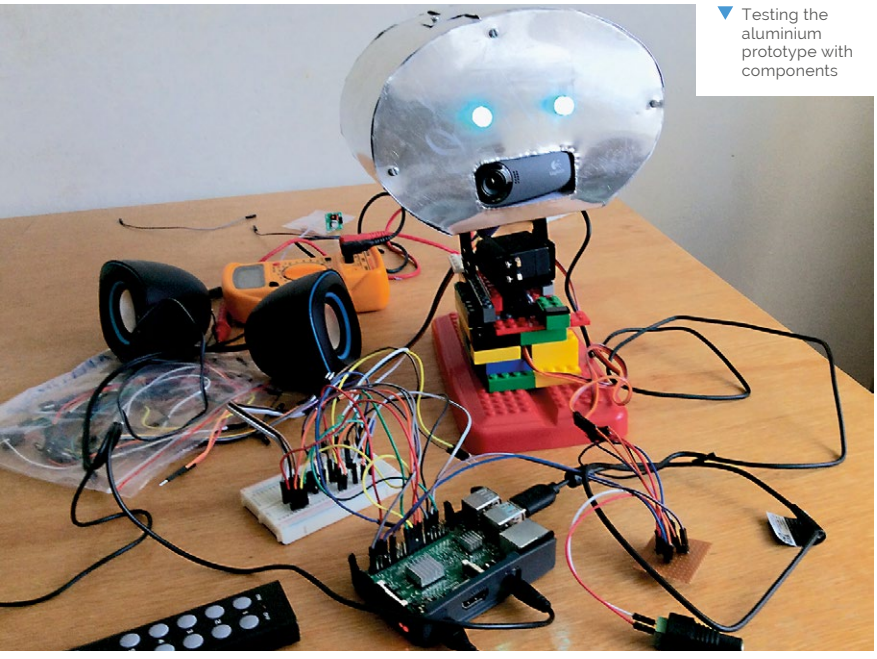
R3-14's head contains a camera for live streaming video and face tracking

Two servos are used with a pan-tilt mechanism to move the head

The robot's body and head are made from 3D-printed PLA parts

Quick FACTS

- Sanjeet worked on the project in the school holidays
- He had no previous electronics experience
- Two servos are used to rotate and tilt R3-14's head
- He wrote a Python library to fade the RGB LED colours
- His next project is a smart headband to control IoT devices



Testing the aluminium prototype with components

accordingly, depending on keywords said, which in R3-14's case would be sending RF signals for the required power socket. "I came across this workaround through experimenting with the imap module," he recalls, "and I developed it into a simple library for others to add Siri voice control to any project in a matter of minutes."

Independent living

While R3-14 is a fun and engaging character to talk to, Sanjeet believes similar robots might be of serious use in the home.

"Robot assistants could one day become the centrepiece in our lives," he says, "and I definitely think they will be useful for giving [elderly people] company, detecting problems in real-time, and in general becoming more helpful: almost like a personal secretary.

"Also, robots like R3-14, if developed further, can give company to those who are alone, whilst allowing them to remain independent and live fuller lives for longer."

The Brixty Four

Usually, bricking an electronic device prompts a flood of tears, but this Lego makeover of a Commodore 64 is sure to make you smile, as **David Crookes** discovers

MAKER

Christian Simpson

Known by the pseudonym Perifractic and based in California, self-confessed retroholic Christian Simpson is a writer, composer, retro-gaming host, and *Star Wars* actor.

magpi.cc/mJSQYR

While not wanting to reheat old playground squabbles, it's hard not to feel that the Commodore 64 is stealing a march when it comes to brick-based recreations. In 2017, the Lego Store in the Arndale Centre, Manchester, showcased a recreated ZX Spectrum home computer. But with no electronic innards, it didn't work – unlike a more recent fully operating Lego Commodore 64 that's on the same level as the real thing.

Created by retro fanatic Christian Simpson, aka Perifractic, the Brixty Four is a joy(stick) to behold. Inspired by a two-inch long Lego C64 produced by Chris McVeigh, it was initially developed as a replacement full-size bread-bin-style case for the 8-bit computer, with the sole intention of putting an actual C64 motherboard and keyboard inside. But then Perifractic's attention shifted to the Raspberry Pi.

Insert Pi

He realised he could slip a Pi into the case and hook it up to a C64 keyboard using an interface called the Keyrah V2b which allows classic Commodore computer keyboards to be connected to modern-day machines. It has proven popular for owners of old 8- and 16-bit computers as diverse as the VIC-20 and the Amiga and it works a treat. "The Keyrah V2b allows the Pi to communicate with the C64 via USB," he explains.

Perifractic also worked on a mechanical keyboard which makes use of a Lego shock absorber from the 1990s Starguider space land-crawler and a set of Lego letter tiles. "I realised I could line four up along an axle to create a line of keys for a fully working sprung mechanical Lego keyboard," he tells us.

From there, he only needed to find a way to emulate the C64, and the solution to that problem soon became apparent. "What could be better to install inside and get it all working than a Raspberry Pi running RetroPie, the best-value C64 emulator by a mile?" he asks. Hooked up to a monitor, Perifractic could soon get down to playing some cool retro games – assuming he could tear his eyes away from the beauty of the case for long enough.

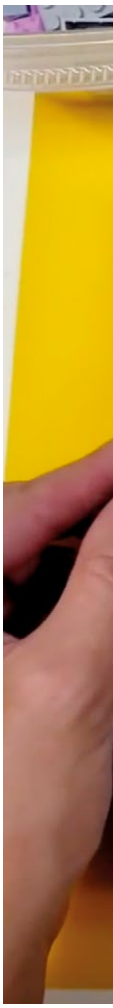
Load Lego

In order to create the case, Perifractic used a freeware app called Bricklink Studio 2.0 which is available for Windows and macOS. "It lets you design Lego masterpieces and play around to see which bricks will work," he says. "To my surprise and delight, I found the smooth, round, front piece to recreate the evocative bull-nose design which is actually from an aircraft hangar set." He ordered all of the pieces he needed from Bricklink (a global brick marketplace) via Studio 2.0 and he soon had everything in place for the build.

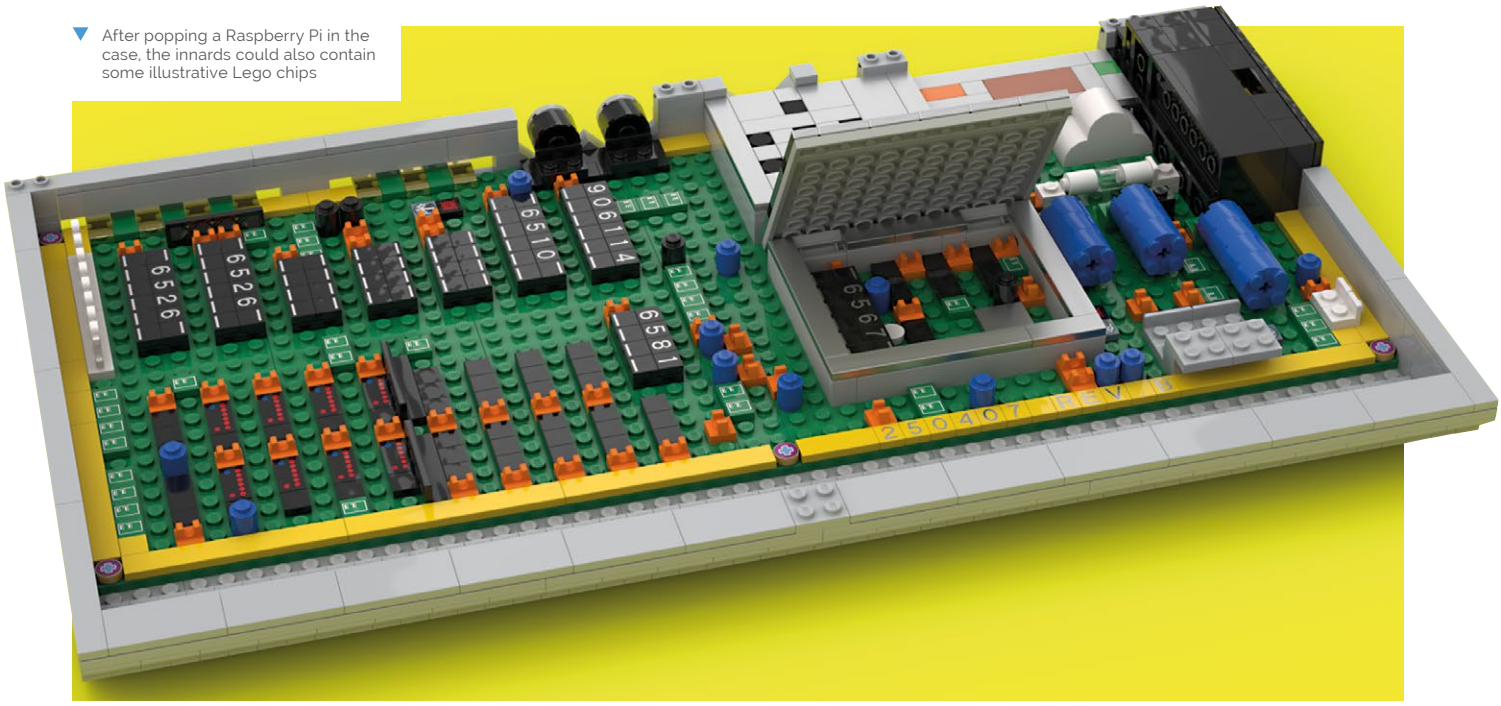
> My ambition is to simply keep these old machines relevant for future generations

"The biggest obstacle was the price – these bricks are rare and used, so to build this machine is costly," he says. "The second obstacle was time, but I wanted to make my final design free and open-source." The results, however, have been more than worth the investment.

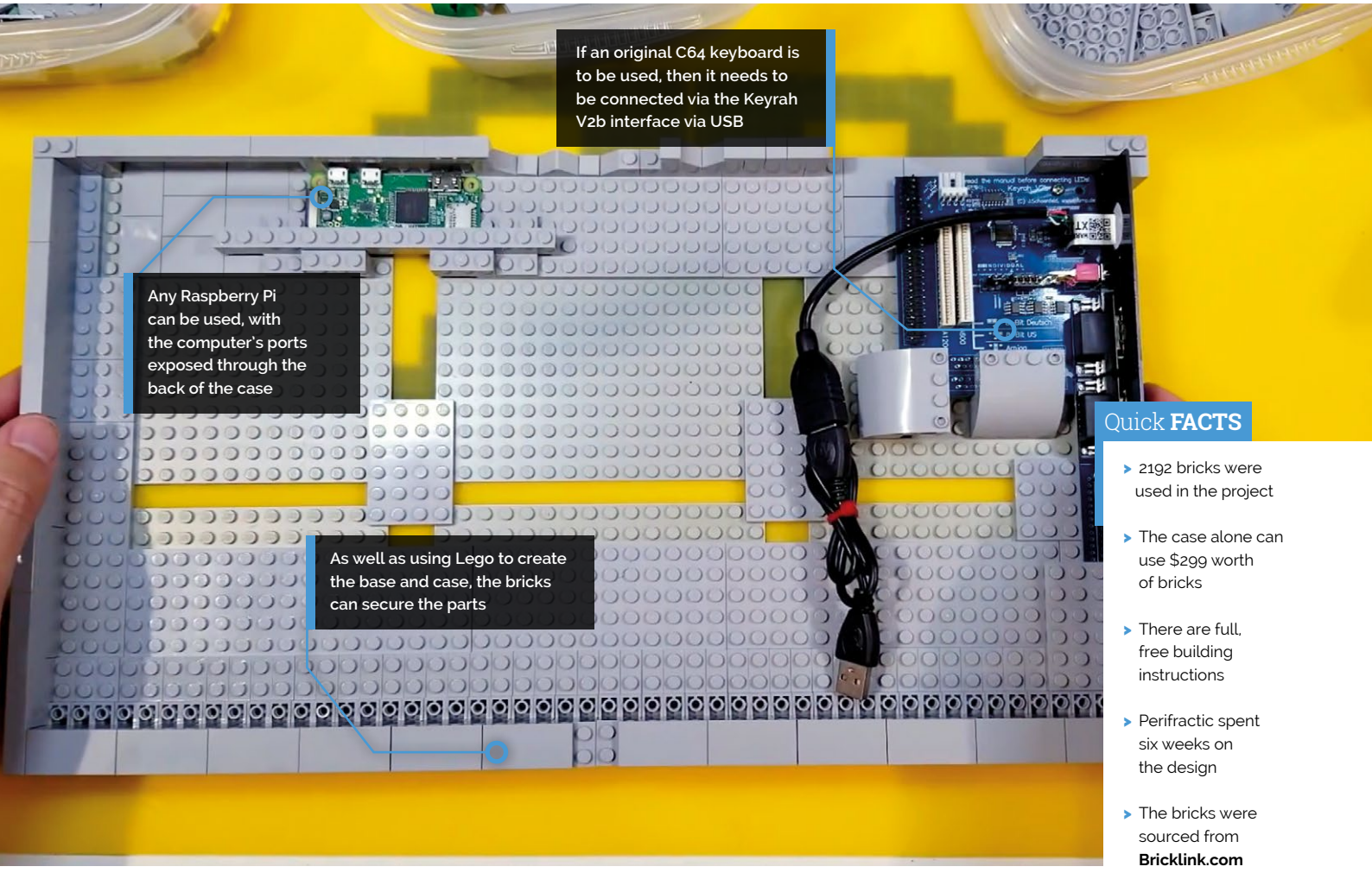
"My ambition is to simply keep these old machines relevant for future generations and I hope it will continue to invigorate discussion," Perifractic adds. "The Raspberry Pi's compatibility and retro-gaming possibilities make it a marvellous modern device running vintage marvels and, with real magic in the old games, it enables anyone to kick back and relax with their favourite game without spending $200 on a real C64."



▼ Thanks to Lego coming in a variety of colours, the case can be far more bold than the original beige



After popping a Raspberry Pi in the case, the innards could also contain some illustrative Lego chips


If an original C64 keyboard is to be used, then it needs to be connected via the Keyrah V2b interface via USB
Any Raspberry Pi can be used, with the computer's ports exposed through the back of the case
As well as using Lego to create the base and case, the bricks can secure the parts

Quick **FACTS**

- 2192 bricks were used in the project
- The case alone can use $299 worth of bricks
- There are full, free building instructions
- Perifractic spent six weeks on the design
- The bricks were sourced from **Bricklink.com**

Raspberry Puitar

What can you do with a Pi Zero and an old electric guitar? **Nicola King** channels her inner Dave Grohl and finds out

MAKER

Behruz Farshi

Software developer Behruz resides in Austria and enjoys tinkering in his spare time, alongside game development, music, and keeping fit.

magpi.cc/fokvqn

When Behruz Farshi picked up his brother's old electric guitar, he began to ponder the endless possibilities if he turned it into a digital instrument. Upon first attempting to play the guitar, he found it very hard work on his fingers: "I was thinking it would be nice to have some kind of instrument that can be played casually, but still is every bit of the real thing," he tells us. "Well, a digital guitar wouldn't need as much tension in the strings and also it could make any sounds you program it to make! So, I ordered a Pi Zero and made a prototype."

Additionally, this was Behruz's first Raspberry Pi–related project, as he explains: "I chose it because it is small, has all the hardware to make it convenient to do something quick, and runs Linux."

Armed with his new Pi Zero, Behruz took around four days to develop both the hardware and the software for his project, with the most time-consuming aspect being the hardware element of the fretboard: "As the idea was to use real guitar parts and space is very limited, it was not easy to assemble it."

▲ Mounted in a recess in the back of the guitar, the Pi Zero plays digital notes through a mini speaker

▶ A rear view of the completed Puitar, showing the Pi Zero enclosed by a piece of Plexiglas

> "As the idea was to use real guitar parts and space is very limited, it was not easy to assemble it"

Musical matrix

The Puitar features a keypad matrix of 22 frets and six strings. When the player presses a string onto a fret, they are connected, which means electricity can flow between them. This is detected on one of the Raspberry Pi's GPIO pins and the appropriate note is played. "To find out which frets are pressed, we loop over the strings, setting [them] high, one at a time and checking which fret turns to show high," explains Behruz. "The rest is a simple software project to make sounds for the pressed frets."

To create the matrix, he drilled holes in the fretboard and soldered wires to the frets from below. After building the first prototype, however, he found

When a string is pressed down on a fret, a circuit is made and detected by an input pin on the Pi, which then plays a note

The Pi outputs electrical current to each guitar string in turn

Quick FACTS

- The Puitar can output MIDI sounds, including a piano
- It's currently monophonic, but a polyphonic version is possible
- The matrix requires more GPIO pins than the Pi Zero has…
- So an IO Pi Plus board was used to expand the number of pins
- Power is supplied by a battery pack with a switch

This standard electric guitar has been modified into a digital instrument

that the frets would sometimes link the strings and make the guitar malfunction. "The solution is very simple: cut each fret in six parts that are separated, and connect them by diodes instead, so they will conduct the electricity only in one direction, and separate the strings, although doing that manually with real fret wire will take ages."

Repeat performance

With this experience behind him, Behruz is now planning a second version of the Puitar: "My plans are a body as close as possible to a real electric guitar (I have already done most of it). And, most importantly, the fretboard with the separated frets – without it, it is not possible to reliably play frets close to each other, as the strings will be connected unintentionally. On the software side, the plan would be to turn the whole project into a MIDI guitar so that I could just connect it over USB to the PC or a smartphone and play any sounds I want."

As it is, this ingenious upgrade of an old electric guitar is music to our ears.

SUBSCRIBE TODAY FROM ONLY £5

SAVE UP TO 35%

Subscriber Benefits

- **FREE Delivery**
  Get it fast and for FREE
- **Exclusive Offers**
  Great gifts, offers, and discounts
- **Great Savings**
  Save up to 35% compared to stores

Rolling Monthly Subscription

- **Low Monthly Cost** (from £5)
- **Cancel at any time**
- **Free delivery to your door**
- **Available worldwide**

Subscribe for 12 Months

£55 (UK)	£90 (USA)
£80 (EU)	£95 (Rest of World)

Free Raspberry Pi 3A+ with 12 Month upfront subscription only (no Raspberry Pi 3A+ with Rolling Monthly Subscription)

Subscribe online: magpi.cc/subscribe

LIMITED OFFER!

FREE!

Raspberry Pi 3A+

WITH YOUR 12 MONTH PRINT SUBSCRIPTION

WORTH £25

This is a limited offer. It replaces our usual offer of a free Pi Zero W.
Offer subject to change or withdrawal at any time.

Buy now: magpi.cc/subscribe

SUBSCRIBE on app stores

From **£2.29**

Available on the App Store

GET IT ON Google Play

ARCADE GAMES

BUILD
A Pinball Machine

CODE
A 3D Game

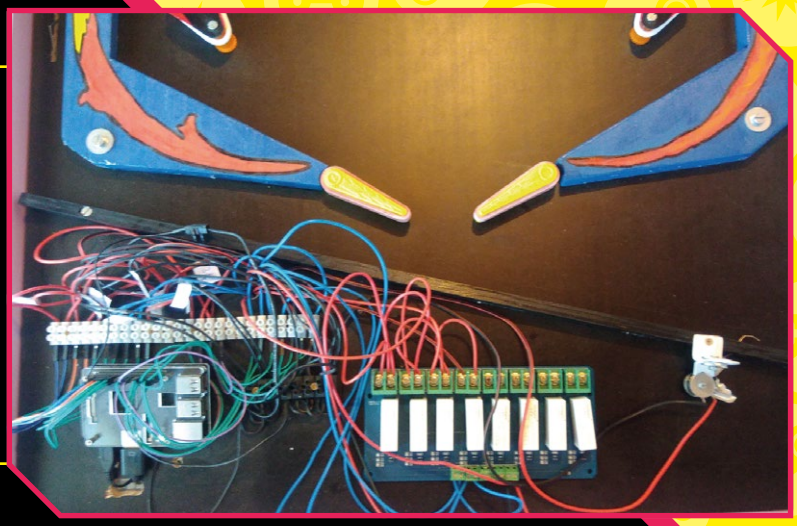
PLAY
PC Games

Try out these fantastic gaming projects with your Raspberry Pi

There are many, many ways that you can use the Raspberry Pi to play games. Whether you're building a physical game, programming your own game, or simply playing games on your Raspberry Pi, it's an amazing piece of hardware for both video and physical games.

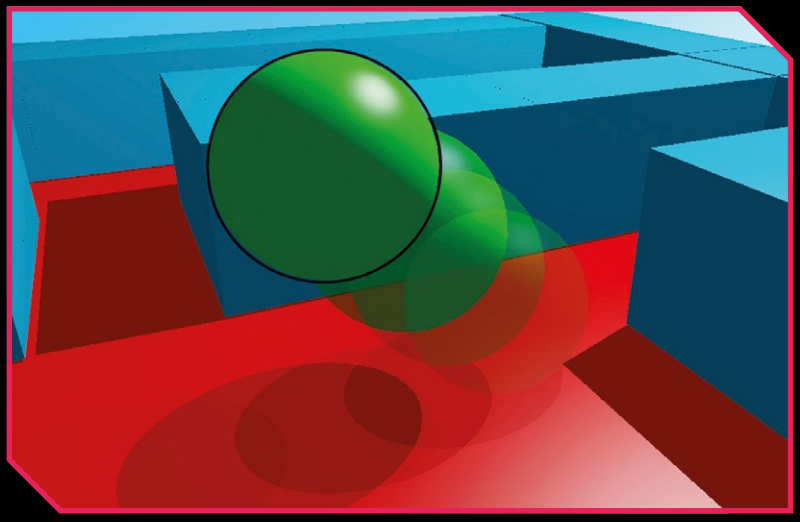
We've brought together three projects that will give you a taste of all three. First, we show you how to build a pinball table out of an old child's bed. Next, play Steam games on your Raspberry Pi. Lastly, code your own game using Pygame Zero.

Let's have some fun.



BUILD A PINBALL TABLE
PAGE 30

PLAY STEAM GAMES
PAGE 38



CODE YOUR OWN GAME
PAGE 42

PINBALL WIZARD

When **Martin Kauss's** kid grew out of her princess bed, he transformed it into an all-singing, all-dancing pinball table. This is how you can do the same

Growing kids leave a trail of detritus in their wake, but it takes a special kind of maker to see an old decorative bed frame and imagine it into the stuff of arcades.

We spoke to Martin Kauss about how he turned his daughter's bed into a one-of-a-kind pinball table, and we've got a step-by-step guide to help you create your own artefact of pinball wizardry.

What possessed you to do this?

I wanted an educational project for myself in terms of GPIO real-world interaction.

In addition, the goal was something inspirational for my kids that would be fun to build and to play with.

As a software developer with a background in mechanics, I wanted to build something real. On top of this, I didn't have much experience in electronics so the project was an introduction to that.

K.G. Orphanides

K.G. is a writer, developer, and maker who's still awestruck at the amazing builds that come out of the Raspberry Pi community.

How long has it taken so far?

In total, I would say about three months. This includes preparation, ordering, reading, learning, building, developing, testing, and optimising stuff.

How much advance planning did you do?

Not so much, to be honest. I made small steps and did a lot of researching, prototyping, and testing upfront for each part.

As soon as I understood one part (e.g. the flipper finger mechanism), I tested the mechanical parts until I got it and then I merged into the software development in combination with the necessary mounting, wiring, and finally the functional testing.

It worked very well for me to focus on one component/part and to move on whenever that part was working.

What was the hardest part?

I had no experience with pinball machines other than playing a few in my childhood, so the hardest part was finding the special vocabulary which leads to sources that helped me understand a component, like the flipper finger mechanism.

The flipper finger mechanic was not easy and, on top of this, dangerous – even with my (in comparison to real pinball machines) low-power ~36V power supply. I burned a resistor in the process. And by burning I mean real flames… albeit very small ones that didn't last long.

What are you planning next on the project?

Even if the Raspberry Pi has lots of GPIO pins I only have very few left. I can only add one more jet bumper or some targets. The decision is not made yet. But the playfield needs some more components, like rubber rings.

Because this project is all about creating a unique pinball table, your parts list won't be identical to Martin Kauss's. However, this parts list will help you work out what kind of components you'll need for your own build, and how much they're likely to cost. Prices have been converted from euros and US dollars to GBP.

The specialist pinball machine components come to around £400, but that's not counting tools, parts for constructing the table itself, and a few extra Pi and GPIO components.

That includes a frame – Martin's daughter's princess bed in this instance; plywood board for the table itself; various sizes of wood and aluminium door trim strip for constructing parts like the shooter alley, outlanes, and the table's legs; a couple of adjusting screws with plastic feet for the table's front legs; and plenty of screws, drive-in nuts, bolts, and mounting brackets to hold everything together.

On the electronics and mechanical components side, Martin says his extra inventory items included LED bulbs, connector strip (aka screw/luster terminal), cable splices, GPIO cables, an extra-tall GPIO stacking header for the Pi, springs, and tension springs (e.g. to close the shooter alley).

Finally, you'll need a reasonably well-equipped toolbox, including a soldering iron and tin lead solder wire, a voltmeter, cable stripper, screwdrivers, saw, a drill and drill bits, a hammer, a hand file, and ideally a helping hand tool with crocodile clips to hold things in place while assembling electronic components.

Number	Component	Price each	Total (GBP)
3	Pinball	£2.67	£8.02
1	Pinball rubber rings kit	£26.91	£26.91
1	Shooter assembly	£26.91	£26.91
2	Cabinet flipper button nut	£1.62	£3.24
2	Cabinet flipper button	£4.50	£9.00
1	Slingshot cover (right)	£8.91	£8.91
1	Slingshot cover (left)	£8.91	£8.91
2	Flipper button switch	£12.42	£24.84
3	Rollover microswitches	£3.78	£11.34
1	Microswitch	£3.78	£3.78
1	Spinner assembly	£25.19	£25.19
1	Full flipper assembly with coil FI-11630, normally open EOS switch (right and left)	£44.10	£44.10
15	Star posts	£0.89	£13.37
2	Pop bumpers	£37.09	£74.18
1	100pc heat-shrink tubing	£4.27	£4.27
1	5m 0.75mm² multicoloured electrical wire	£11.07	£11.07
1	36V power supply (TDK-Lamda LS150-36)	£39.30	£39.30
1	20cm male-female jumper cables × 40	£4.73	£4.73
1	2m LED light strip	£8.09	£8.09
1	5V power supply (TOOGOO(R) AC 110V/220V)	£11.57	£11.57
1	SainSmart 8-channel 5V solid-state relay module	£12.63	£12.63
1	SainSmart 2-channel DC-DC 5V-220V 5A solid-state relay	£11.05	£11.05

TOTAL: £391.41

Useful Retailers

- flipperteile.de
- flipperservice.de
- pinballlife.com
- sainsmart.com
- pinball.co.uk
- pinparts.co.uk

MAKE YOUR OWN PINBALL MACHINE

This step-by-step guide breaks down the key stages of building the Princess Pinball table. While your table could be very different, the key components and techniques apply to a wide range of builds.

Every pinball table will need a shooter, flippers, bumpers, and rubbers. And tips – like using adjustable legs to help you get the perfect angle on your table – hold true across many different build styles.

Similarly, Martin Kauss's GPIO connection configuration and software to run your table's lights, sensors, sound, and scoring are powerful tools for any pinball build.

01 Set up the software

Start with a Pi and a clean install of Raspbian. You'll need an internet connection, and your life will be easier if you connect a monitor, which we'll also use later to track the player's score, keyboard, and mouse.

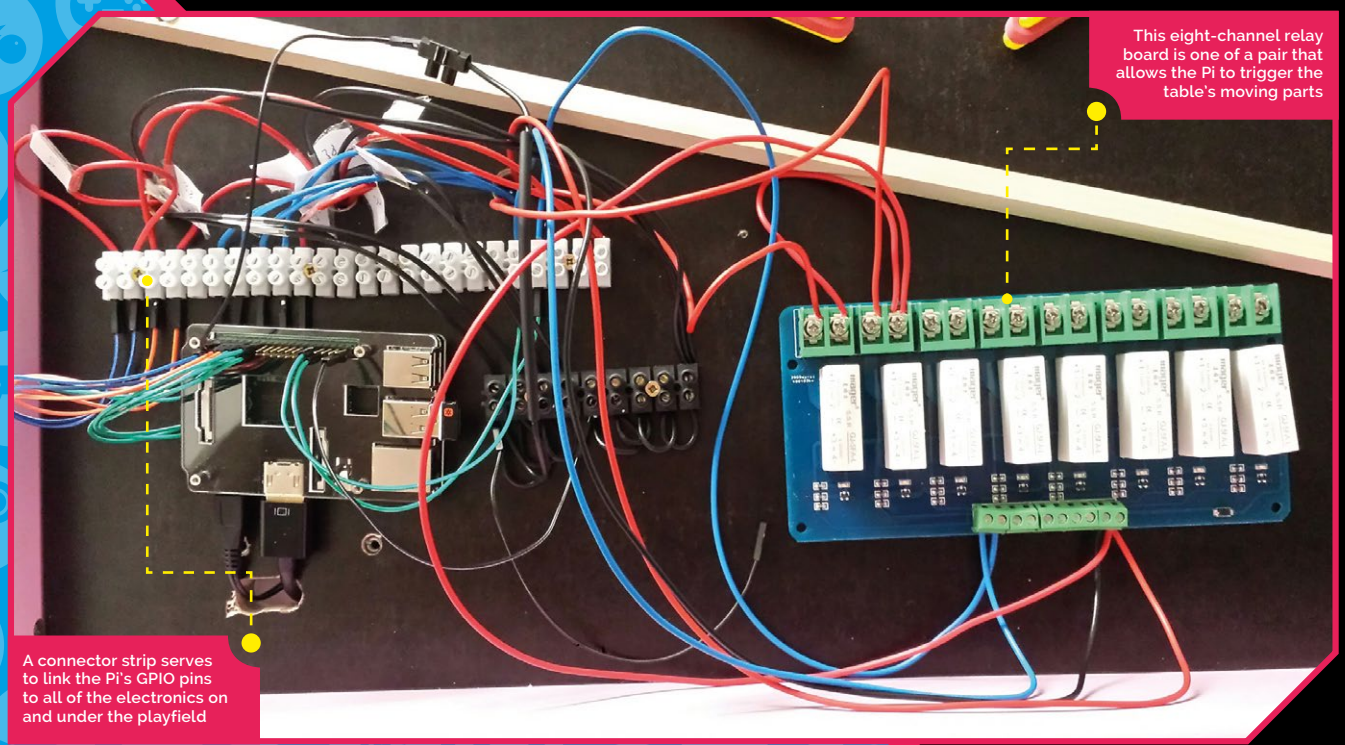
Open a Terminal window and type:

```
sudo apt install python-pygame
git clone https://github.com/bishoph/pinball.git
cd pinball
python pinball_machine.py
```

Pressing **Q** exits the program. Fonts aren't included, so to run the program you'll need to find your own **pinball.ttf** and **comicfx.ttf** TrueType files and copy them into **/usr/local/share/fonts/** – both are freeware and available online.

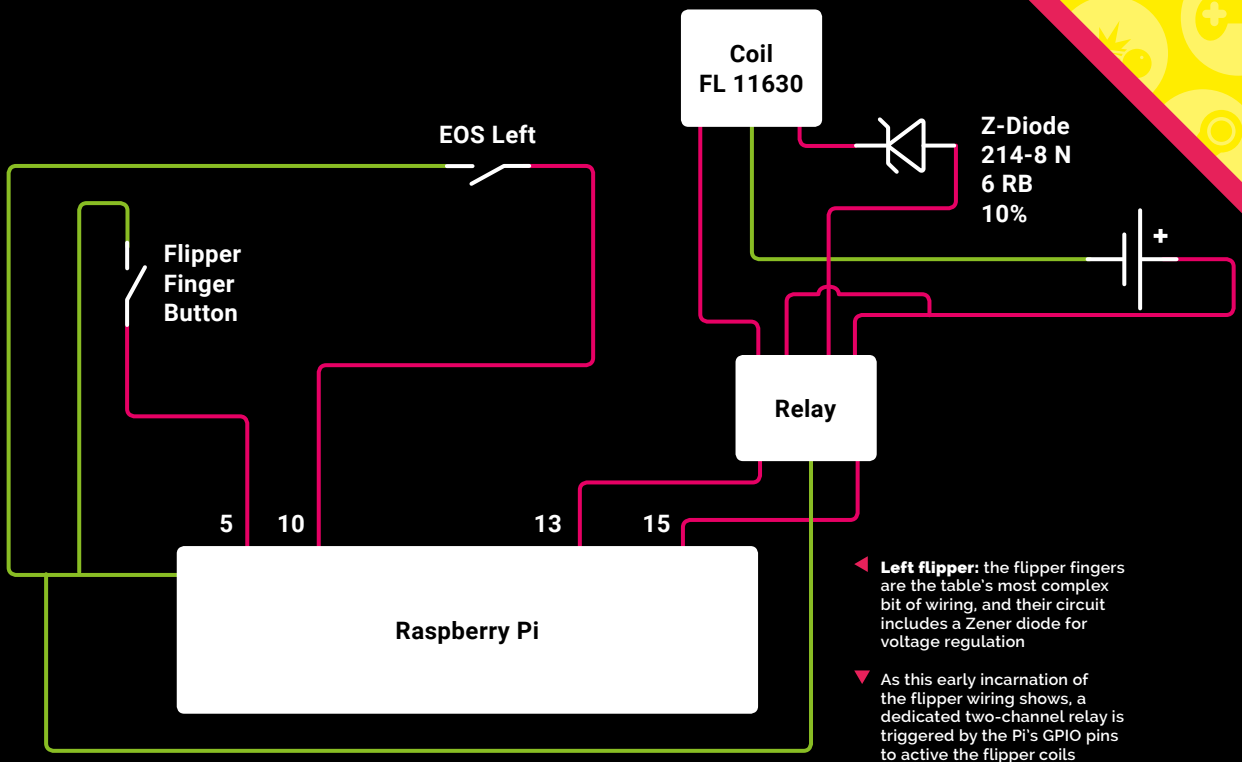
02 Vision on

Setting everything up is much easier if you've got a monitor connected to your Pi, but



This eight-channel relay board is one of a pair that allows the Pi to trigger the table's moving parts

A connector strip serves to link the Pi's GPIO pins to all of the electronics on and under the playfield

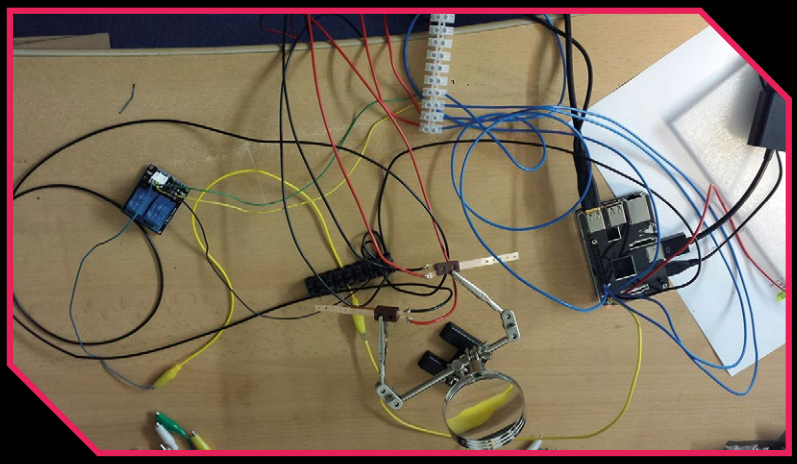


Left flipper: the flipper fingers are the table's most complex bit of wiring, and their circuit includes a Zener diode for voltage regulation

As this early incarnation of the flipper wiring shows, a dedicated two-channel relay is triggered by the Pi's GPIO pins to active the flipper coils

our pinball table's Python scripts can also make use of an external display. We'll use this to show the player's current score, the number of balls left to play, the table's high score, and a few fun visual effects.

If you're feeling really sharp, you could use a VESA mount to affix the monitor to a backboard or stand attached to the table; or, if you're working with a bed frame like Martin, find space to attach it to the one-time head of the bed.



03 Face the music

You can play sounds either through the integrated speaker of a monitor or by connecting speakers to the Pi's 3.5 mm audio output. Sounds are defined in the **effects.py** script, which we installed in Step 01. You'll have to source your own audio files – Martin got some from **freesfx.co.uk**.

Place them in the **/home/pi/pinball/sounds** directory and edit **effects.py** accordingly. The script triggers sounds when the table is powered up, when your ball heads down the shooter alley, when it falls out of play by going down the outlane, and when it triggers the spinner or pop bumpers.

Samples in a bank called s2 are triggered as random events when the table is idle.

04 Frame and fortune

Martin Kauss's pinball table began life as a child's bed, decorated with colourful Disney princess imagery, but you could also build your own frame out of wood, buy a table base kit, or even make one out of K'nex or Meccano.

The frame measures 145×77 cm and for the playfield – the surface of the pinball table, where all the action happens – Martin used a piece of 230 mm thick multiplex board (plywood) with a black finish.

Beneath the playfield, you'll need space for your wiring and power supplies.

Test first

Make sure your components and connections work as intended before you permanently screw them into place.

05 More power, Igor!

This build calls for both a 5 V PSU, which handles the lights, and a 36 V one for the coils used to power the flippers and bumpers, which have comparatively high voltage requirements.

For example, although the pop bumper's coil is connected to the 36 V supply, its built-in LED light, like those elsewhere on the table, is powered by the 5 V PSU. For ease of connection, a GPIO stacking header is mounted on the Pi's GPIO.

From that, GPIO cables run to a connector strip. That includes both inputs and outputs, and all powered components such as lights and coils are wired up via relay boards. The relays are powered from the Raspberry Pi power supply (via physical pins 2 and 6) and the trigger action for each comes from the respective output GPIO pin.

It's useful to screw a small four-way plug bar to the rear or underside of your table, as you'll need separate power supplies for your monitor and the Pi.

06 Tilt! Tilt! Tilt!

Pinball tables can't be flat, but getting the correct angle to ensure that the ball rolls towards the bottom at the right speed can be tricky. Adjustable legs mean that you can easily set and change the table's incline.

In this case, four square wooden battens measuring 4.5×4.5×100 cm at the front and 4.5×4.5×110 cm for the rear have been used to make front and rear legs. The front pair have been made adjustable by using screw-in plastic feet which can be raised or lowered, allowing for a bit of trial-and-error engineering during construction.

07 Plan, sketch & drill

Once you've got your frame and a table surface to fit it, it's time to outline its features. Carefully measure, test-place, photograph, and draw around the flippers, shooter, slingshot, and

▲ **Spinner:** the spinner itself is passive, but a microswitch is activated every time the ball flips it, triggering scoring, sound, and lights

▲ You'll need a healthy selection of tools, electronic components, and other parts for a project like this, as well as the space to build it

bumpers you want to include. You'll also want to place side-mounted flipper buttons.

As well as these components, you'll need to use wood and metal strips for your lanes, an outlane area to direct the ball back to the shooter alley when it's lost, and the curved upper part of the table. Once you've marked up and double-checked, drill the holes you'll need to bolt on and wire up your components.

> "Remember to leave enough space at the bottom of the table for your electronics"

Remember to leave enough space at the bottom of the table for your electronics, including the Pi!

08 Assemble the plunger

The plunger, also known as the ball shooter, is the first critical bit of your table to assemble. You can buy the whole assembly as a kit, including a rod, springs, housing, and mounting components such as the external trim plate.

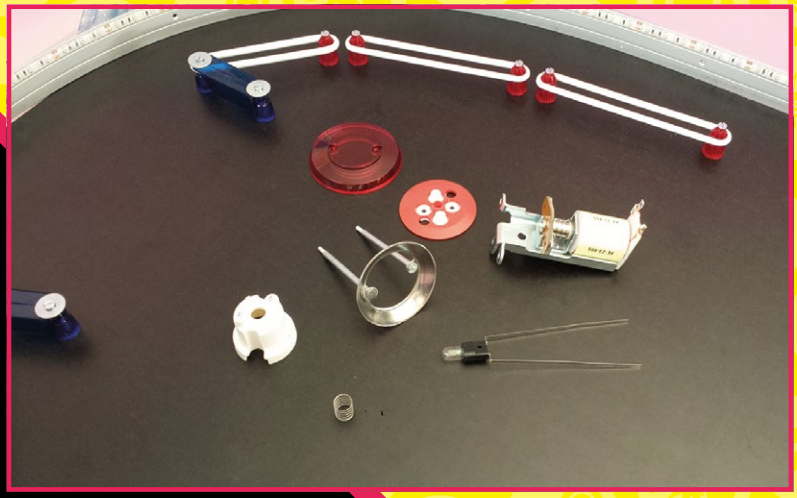
You'll want to put it at the left of your table, with the knob and external parts protruding from the front of your pinball machine. A length of wooden batten forms the lane that the ball will travel down, and the drain beneath the flippers should direct lost balls back to the shooter assembly. A high-tension spring holds a lane closure flap shut until it's forced open by the velocity of the ball coming through from the shooter.

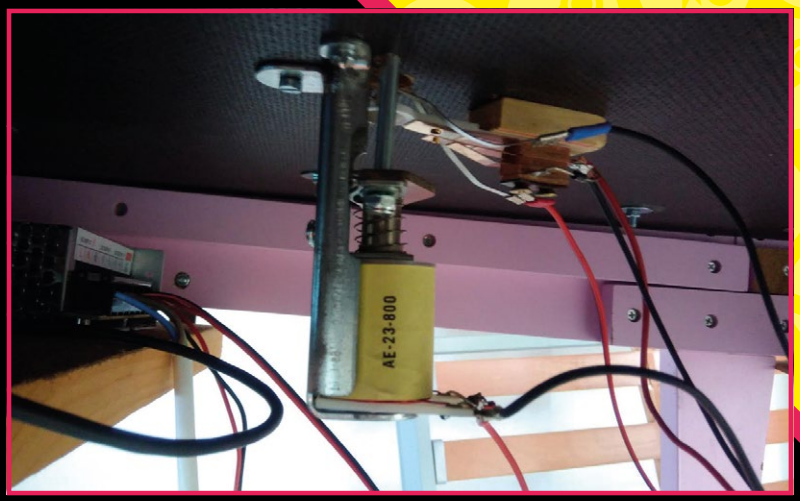
09 They see me rollin'

Most of the lanes and structural parts of this design are made from wood or aluminium strip bent into shape and held in place using wooden blocks screwed into the table, all of which makes for a pretty forgiving build. But you'll want the Pi to be able to tell when the ball passes through those lanes.

For that, you'll need some rollover microswitches from a pinball part supplier, which you can fit to the table from below. Cut a channel in the playfield using either a router or a drill and jigsaw. Mount the switch on a strip of wood, and position it beneath the channel so that the switch protrudes enough to be triggered by the weight of the ball as it passes.

This build has rollover switch at the end of the shooter alley and on all the outlanes that take





▲ The pop bumper is assembled with its coil below the table and its skirt and cap above

Go online for more details

For extra information and a maker's blog about the Princess Pinball table, check out Martin Kauss's website at magpi.cc/iimYKq

the ball out of play, plus one at the very end, just before the ball re-enters the shooter alley, so the table knows when the ball leaves the playfield.

They're all connected to the Pi's GPIO header – via a connector strip – so they can trigger events such as lights sound and scoring.

10 Flip the finger

A pinball table's flippers are its most important components. You'll need a 36 V, 5 A PSU to give the flipper coils enough of a kick, and a two-channel relay to activate them when triggered by the Pi's GPIO.

Because we're using a Pi to control everything, we'll need a modern flipper assembly with a normally open (NO) end-of-stroke (EOS) switch.

The coils are mounted beneath the table's flipper fingers. For each flipper, you'll need to connect a button on the side of the cabinet to one GPIO pin, the EOS switch to another, and then another two pins – via the relay – to the flipper coil units, which actually have two different wire coils wrapped around them.

The first coil, HIGH, provides low resistance and makes the flipper finger move hard and fast, while the second, HOLD, has high resistance and allows you to hold down the flipper buttons to keep them upright.

Pin	Connector Strip Terminal	Type	Description	Relay Connection
3	1	IN	Flipper button right	
5	2	IN	Flipper button left	
8	3	IN	Flipper finger EOS right	
10	4	IN	Flipper finger EOS left	
7	5	IN	Spinner microswitch	
11	6	OUT	Flipper finger right HIGH	Relay #1,1
12	7	OUT	Flipper finger right HOLD	Relay #1,2
13	8	OUT	Flipper finger left HIGH	Relay #2,1
15	9	OUT	Flipper finger left HOLD	Relay #2,2
16	10	IN	Shooter alley microswitch	
18	11	IN		
19	12	IN		
21	13	IN		
22	14	IN	Bumper #1 switch	
23	15	IN	Bumper #2 switch	
2	16	IN	Slingshot switch	
26	17	OUT		
29	18	OUT		
31	19	OUT		
32	20	OUT	Light #1, shooter alley	Relay #3,1
33	21	OUT	Light #2, slingshots	Relay #3,2
35	22	OUT	Light #3, bumper	Relay #3,3
36	23	OUT	Light #4, bumper	Relay #3,4
37	24	IN	Outlane microswitches, one signal!	
38	25	IN	Bumper #1, coil	Relay #4,1
40	26	IN	Bumper #2, coil	Relay #4,2

The Princess Pinball table has proved to me a massive hit with Martin's kids and all their friends, complete with requests for a future multi-ball feature

11 Into the slingshot

Slingshot bumpers are the wedge-shaped components positioned just above your flippers, helping to bounce your ball back to them on its journey around the table.

They're made up of a set of star posts – brightly coloured plastic mounts that a rubber ring can fit around and a rubber surround for the ball to bounce off. A microswitch sensor detects when the ball hits, racking up points, and triggering noises and lights.

A pair of fancy slingshot covers rounds out the look, and a lane – made of wood – runs below them to provide a route to the flippers.

12 Add a bumper or two

Mushroom-shaped pop bumpers are among the most iconic elements of a pinball table. They're available in complete kits and most work by detecting, via an integrated microswitch, when the ball hits their plastic skirt. A coil then pulls down the bumper's rod and ring assembly to kick the ball away.

You can use the bumper's base to mark out where you want to mount it, with the coil mounted below the playfield surface. The coils are powered by the 36 V PSU and connected to the Pi via the eight-channel relay.

13 Add a spinner, connected via screw terminals

A spinner, or spinning target, is a classic pinball table component with its own lane and a microswitch that triggers lights and a score increment when the player hits it.

Here, the lane is created using a metal rail of aluminium threshold strip. The spinner is mounted using some DIY-shop metal brackets and bolts, plus star posts and rubber rings from a pinball supply firm.

A straight-wire actuator microswitch is connected to the spinner to detect the ball's passage and rack up points and flash lights around the table in response.

14 Lights! Action!

A pinball table is nothing without flashing lights. This project uses a 2 m LED light strip connected to the 5 V PSU to add some pizzazz to the table's shooter alley and orbit – the upper loop of aluminium strip that helps keep the ball rolling around the table.

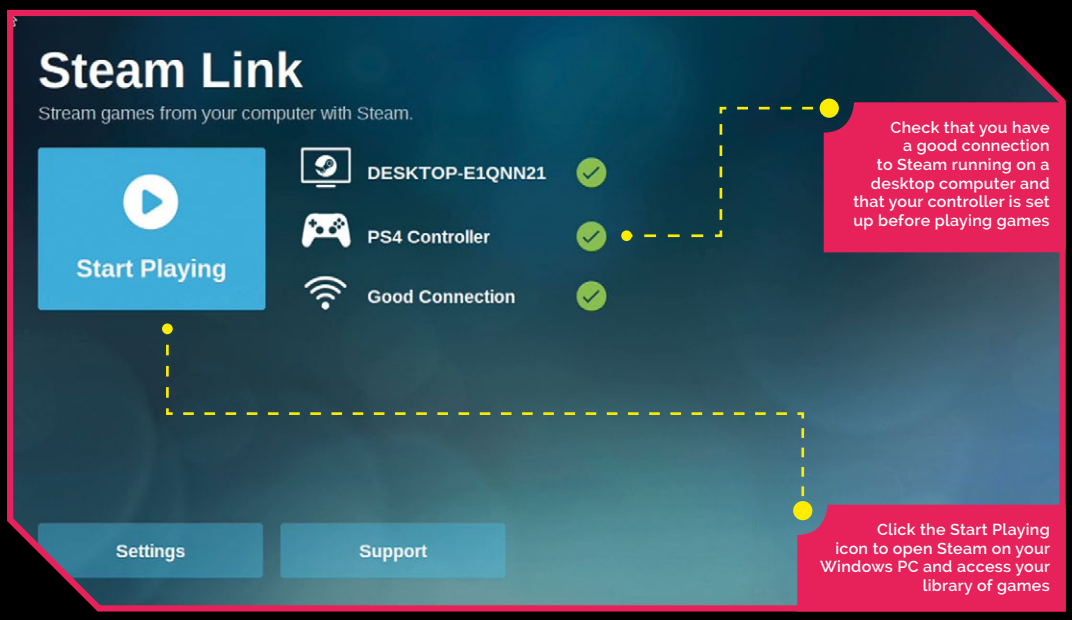
There are also individual lights in each of two bumpers and the slingshot bumpers. That's a total of four GPIO connections on the Pi. The Python script we're running will activate the lights when the ball enters the playfield, hits bumpers, triggers the spinner, or passes over rollover switches. The lights are also triggered by the script at random events if there is no input or action on the table.

Fingers to yourself

Don't touch the coils or moving parts during testing, as the rapid contraction of the solenoid can deliver a painful pinch.

PLAY PC GAMES ON A RASPBERRY PI

Turn your Raspberry Pi into a Steam Link machine and stream full-quality games from Steam on a PC


Steam Link
Stream games from your computer with Steam.
Start Playing
DESKTOP-E1QNN21
PS4 Controller
Good Connection
Settings
Support
Check that you have a good connection to Steam running on a desktop computer and that your controller is set up before playing games
Click the Start Playing icon to open Steam on your Windows PC and access your library of games

You'll Need

- Raspberry Pi 3B+/3B
- Raspbian Stretch
- Keyboard and mouse for setup
- Windows PC
- Steam (and Steam account)
- Compatible controller (see boxout)

Steam is an online store packed with all the latest PC games, and it recently announced Steam Link support for Raspberry Pi.

With Steam Link, you can stream Windows games from a PC computer to your Raspberry Pi. Hook up a gamepad and you have a games machine that'll rival any console.

You can use just about any gamepad with your Raspberry Pi (see 'Compatible Controllers') and you can play full-quality PC games on any television or monitor around the home.

It works alongside the Steam software on a Windows PC, so you'll need a Steam Store account and a game from the Steam store (there are plenty of free games and demos to play).

Let's take a look at how to turn a Raspberry Pi into a Steam Link-powered console and play Windows games on it.

Lucy Hattersley

Lucy is editor of *The MagPi* and loves RPG and adventure games. She was really surprised at how enjoyable Euro Truck Simulator 2 was, and will drive a virtual truck around Cambridge any time.

magpi.cc

01 Set up Steam

Start by opening and logging in to Steam on your Windows PC. If you don't have Steam installed, you can download it from **store.steampowered.com**. Click Install Steam

to download the **SteamSetup.exe** file into your Downloads folder. Double-click it and follow the Steam Setup process. Click on Create a New Account and follow the account-creation process. Finally, log in to the program.

You'll need a game to play. Fortunately, Steam has a great selection of free software and demo games to try out. Click on Games and choose Free to Play to see a range. We chose to install the Demo of Euro Truck Simulator 2.

02 Install Steam Link

With Steam up and running on your Windows PC, it's time to turn your Raspberry Pi into a Steam streaming box. Start with a fresh installation of the Raspbian Stretch software (see **magpi.cc/quickstart** if you need a refresher on how to install Raspbian). Make sure your Raspberry Pi is connected to the internet (ideally with an Ethernet connection) and open a Terminal window. Enter the following commands:

```
sudo apt update
sudo apt install steamlink
```

Answer 'y' to any questions.

03 Connecting a controller

We're going to connect a PlayStation DualShock 4 gamepad to the Raspberry Pi using Bluetooth.

You can use a wide range of other controllers, including Xbox One, Nintendo Switch Pro Controller, and the Steam Controller (along with many other Bluetooth gamepads). Or you can stick with a keyboard and mouse for many games. But a controller is best if you want to use your Raspberry Pi with a television.

Keep your keyboard or mouse connected to the Raspberry Pi while you set things up. Click on the Bluetooth icon in the Menu bar and choose Turn on Bluetooth. Now click on the Bluetooth icon again and choose Add Device.

Put the gamepad into pairing mode. The process for this is different on each gamepad. On a DualShock 4 Controller, you hold down the PS4 and Stream button on the gamepad until the light on the back starts flashing.

You should see Wireless Controller in the Add New Device window on your Raspberry Pi. Select it and click Pair. You should see 'Pairing Successful'; if not, try clicking on Bluetooth > Wireless Controller > Connect.

It took us a few attempts to get the PS4 gamepad to connect to the Raspberry Pi. Keep at it.

▲ Driving around our Cambridge base in Euro Truck Simulator 2, running on a Raspberry Pi

04 Start Steam

Click on Menu > Games > Steam Link to open the app (you can now close Terminal).

The Steam Link application will open in full-screen mode. You should see three things: the name of your computer (in our case 'DESKTOP-E1QNN21'), 'PS4 Controller' (or the name of your controller)', and 'Good Connection'.

First, let's test the connection. Use the gamepad to click on Settings and choose 'Streaming and Network test'. The Testing Network window will appear. Hopefully, you will see 'Network Test Complete' and 'Fantastic! Your network looks like it will work well with Steam Link'. Steam recommends using an Ethernet connection, but if you're attempting to use WiFi you can try moving your Pi closer to the router. Click OK and use the Back icon in the top-left to head back to the main window.

05 Start playing

Click the Start Playing icon in Steam Link. You should see a four-digit PIN code in Steam Link on your Raspberry Pi. Enter the PIN in the Authorize Device dialog window in Steam on your Windows PC and click OK.

At some point you may see a 'Can't connect' warning dialog along with a message that

Ethernet connection

Steam recommends you use a wired Ethernet connection to connect your Raspberry Pi to your network router. We found streaming games to be fine, but some of the menus were a little sluggish. Hopefully, performance will improve over time.

Steam Link Forum

Be sure to join the Steam Link forum to get help with any of your questions. There's a whole section on Raspberry Pi and troubleshooting. magpi.cc/xJopz0

'Streaming requires additional driver installation'. Click on Install to download and install the drivers required. Click on Start Playing again to enter Steam Link.

06 Welcome to Steam

You will now see the Steam Link 'Welcome to Steam' windows. Along the top will be a strip of icons: Web, Store, Library, Community, and Chat. Above that are three smaller icons for Downloads, Settings, and Power. You should be able to use the D-pad on the PS4 gamepad to navigate around the icons. Press the X button to select items and O to go back.

07 Controller Settings

Basic support for the gamepad is working, but for better support you need to set up the controller. Click on the Settings icon in the top-right and choose Controller Settings. Move down to PlayStation Configuration Support and click again to place a tick in the box. The 'Personalise Your Controller' window will appear. Leave the default settings for now. Choose Submit. Press O to go back to the main window.

08 View Library

Select Library in the Welcome to Steam window to view all the games in your Steam collection. Choose Installed and Euro Truck Simulator 2 Demo. You will see a yellow warning that says 'Controller Configuration Required'. Even though your PS4 gamepad is set up in Steam, it may require configuration for each individual game (many Steam games are designed with a full keyboard and mouse in mind). We'll come to this next; for now, click Play.

09 Config the controller

As this is your first launch, the First Time Config Selection will appear. Click Browse Other Configs. Choose Community and you will see a bunch of bindings created by other people. To the right they will have a number of Votes. Pick the one with the highest. We see 'zaka.ahoa2's Bindings' with four upvotes. Press the Square button to Apply Configuration and start the game.

10 Pick your own controls

You can now play the game using the PS4 trackpad to start a driving job. When you start to drive, the D-pad or left analogue stick will control movement, and you can look around the truck using

▼ The Steam interface is reconfigured to work on a television screen with a controller

▲ Even with the Controller connected to Steam Link, you will receive warnings to configure it (to personalise the controls) for many games

the right analogue stick. Take your truck for a drive (or play whichever game you are on). While it's best to start with community-created controls, you can add your own controls to each game and personalise them according to your own preferences.

11 Custom controls

Press the PlayStation button to access the Settings window. Here you can see Controller Configuration. In this screen. You can set custom controllers. Select the left shoulder button and use the virtual keyboard to set it to '['; then set the right shoulder button to ']'. Now, as you drive, you can use the left and right shoulder buttons to use your indicators.

Many PC games have complex controls, and you may not be able to play all games using just a gamepad. But many games are perfectly playable using just a controller. Have fun playing PC games on your Raspberry Pi.

12 Play the game

Now you can go ahead and play your game. Make sure the controls all work correctly and you should be able to move your Raspberry Pi from its test environment to a place underneath your television. A quick way to do this is to add it to the **.bash_aliases** file. Open a Terminal window and enter (don't forget the '.' before 'bash':

```
sudo nano .bash_aliases
```

This should open a blank file. Enter:

```
steamlink
```

Press **CTRL+W** to save the file, and **CTRL+X** to exit the Nano text editor. Now restart your Raspberry Pi (`sudo shutdown -r now`). When it boots, it should launch directly to Steam Link.

COMPATIBLE CONTROLLERS

Use these gamepads to play games. For a complete list of supported adapters and features, see this list: **magpi.cc/BoqAxE**

PS4 DualShock 4

The PS4 DualShock 4 Controller (£50) is a popular choice, and the pad we use in our tutorial. As well as a D-pad and two analogue sticks, it features a touchpad that can replicate mouse input.

Xbox Controller

The Xbox Controller (£50) works wirelessly using the same setup as the PS4 controller. Although it isn't quite as well supported as the DualShock 4, you will be able to play games with it.

Nintendo Switch Pro Controller

It's an expensive choice but if you have one to hand, the Nintendo Switch Controller (£60) works well with Steam Link. It lacks the touchpad of the DualShock 4, though.

Steam Controller

Steam sold its own unique controller alongside the official Steam Machines (which the Raspberry Pi now replaces). You can still pick up a Steam Controller (£40). Alongside an analogue stick are two trackpads designed to replicate mouse input. magpi.cc/oBzbjt

Part 01

CODE AN ISOMETRIC ADVENTURE GAME: AMAZEBALLS

Pygame Zero in 3D. Yes it's possible and we will show you how in this three-part maze game tutorial

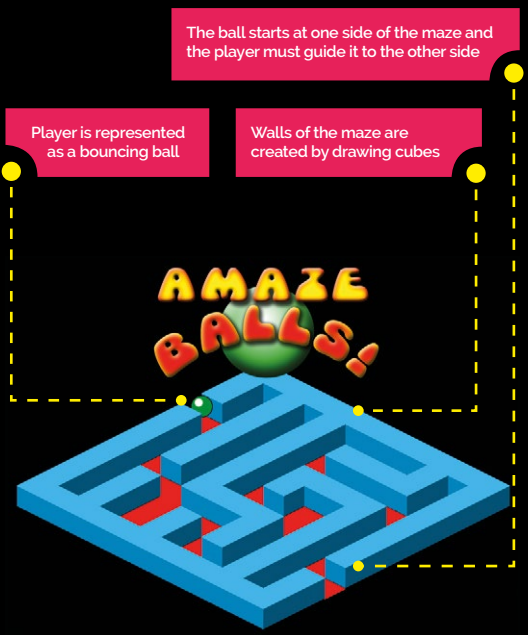
In this series so far, we have learnt how to use Pygame Zero to quickly create games. In this three-part tutorial, we will use several more new techniques to create a 3D maze game. The style of 3D graphics we'll be using is called isometric. That means that our display will be made of regular cubes that have a slightly false perspective because the display is actually made of 2D images. This technique was used in many eighties games such as Knight Lore, Alien 8, and my own series, ArcVenture. In this first part, we'll build the basic map using data lists, and create a bouncing ball character for the player to move around the maze.

01 Welcome to the third dimension

Pygame Zero was not written with 3D games in mind, but sometimes you just have to push the boundaries a little and see how far you can take an idea. This method of drawing a game area is not strictly speaking a 3D display, but it does look 3D and we can do a lot with this technique. We are going to make a map from list data and build it up out of cubes. We will then put a bouncing ball into the game area for the player to move around using the keyboard. We'll start the ball at one side of the maze and when the player guides it to the other side, the game is complete.

02 Map-making

As usual in this series, we start with importing our `pgzrun` module and don't forget


The ball starts at one side of the maze and the player must guide it to the other side
Player is represented as a bouncing ball
Walls of the maze are created by drawing cubes
AMAZE BALLS!

Mark Vanstone

Educational software author from the nineties, author of the ArcVenture series, disappeared into the corporate software wasteland. Rescued by the Raspberry Pi!

magpi.cc/YiZnxL | @mindexplorers

to call `pgzrun.go()` right at the end of the code. The default window size should suit our purposes. We are going to make a two-dimensional list of numbers that will represent our map. Each number will represent one square on the map and we will make our map twelve squares wide and twelve squares high. See **figure1.py** to see how we define this list, which we will call `mapData`. You will also see another variable called `mapInfo` that defines the width and height of our map in a structure called a dictionary.

03 A is for aardvark

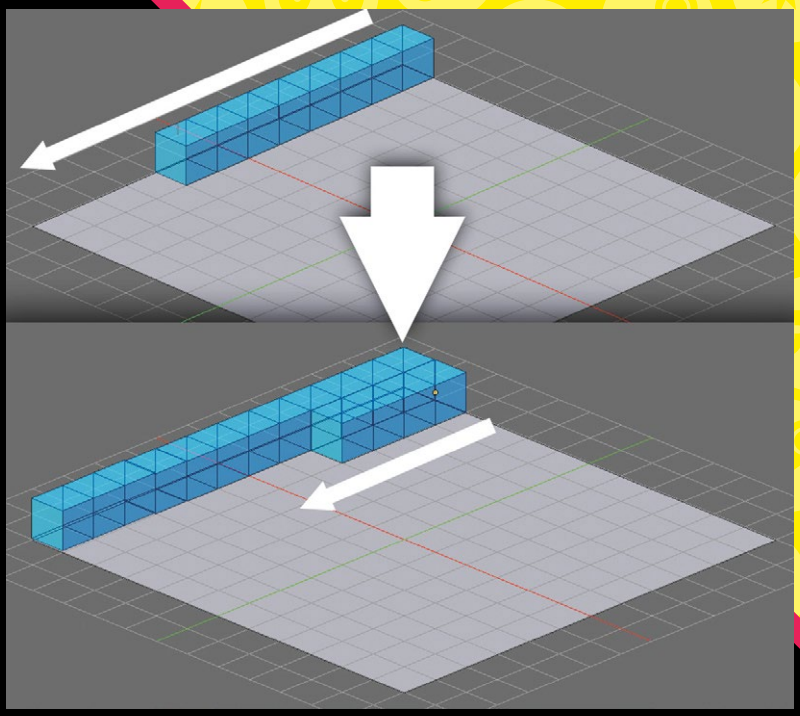
A dictionary in Python is a very useful data structure. Rather than just storing a list of values or strings, we can give each of these values a label. Looking at our `mapInfo` dictionary, we can see that there are two values declared; the first has the label 'width' and the second 'height'. To read the values back, we just need to write `mapInfo["width"]`, which would give us the value 12 in this case. Dictionaries are a very useful structure to gather together several variables that are all associated and can be referred to by their labels.

04 Mapping the map

Now that we have some data for our map, we need to define some images that are going to be used to draw the map. We can make a new list of map blocks by writing `mapBlocks = ["map1c","map2c"]`. What this will do is define that when we see a 0 in the `mapData` list, it means draw the first image in the list, which is **map1c**. If we see a 1 in the data, then use image **map2c**. For the moment, we'll just stick to two different map blocks. The first will represent the floor; the second, the walls.

05 Show me the map

Next is to get our map data translated into a visual map we can see on the screen. We'll set up the basics in our Pygame Zero `draw()` function and then call a `drawMap()` function that we'll write to do the work. In **figure2.py**, you'll see that we fill the screen with black first, then draw our map. The `drawMap()` function, although short, may look a bit complicated, so let's go through it slowly as you'll need to understand what is happening here.



▲ The blocks build back to front, one row at a time, to produce the 3D effect

▼ Our basic Pygame Zero framework and our map data definitions

figure1.py

DOWNLOAD THE FULL CODE:

magpi.cc/DVNCvv

Language: **Python**

```
001. import pgzrun
002.
003. mapData = [[1,1,1,0,1,1,1,1,1,1,1,1],
004.            [1,0,0,0,0,0,0,0,0,0,0,1],
005.            [1,1,1,1,1,1,1,0,1,1,1,1],
006.            [1,0,0,0,0,0,0,0,0,0,0,1],
007.            [1,1,1,1,1,1,1,1,0,0,0,1],
008.            [1,0,0,0,0,0,0,1,0,1,1,1],
009.            [1,0,1,0,1,1,0,1,0,0,0,1],
010.            [1,0,1,0,1,0,0,1,1,1,0,1],
011.            [1,0,1,0,1,0,0,0,0,0,0,1],
012.            [1,1,1,0,1,1,1,1,1,1,1,1],
013.            [1,0,0,0,0,0,0,0,0,0,0,1],
014.            [1,1,1,1,1,1,1,1,0,1,1,1]]
015.
016. mapInfo = {"width":12, "height":12}
017.
018. pgzrun.go()
019.
```

figure2.py

```
001.  OFFSETX = 368
002.  OFFSETY = 200
003.  mapBlocks = ["map1c","map2c"]
004.  mapHeight = [0,32]
005.
006.  def draw(): # Pygame Zero draw function
007.      screen.fill((0, 0, 0))
008.      drawMap()
009.
010.  def drawMap():
011.      for x in range(0, 12):
012.          for y in range(0, 12):
013.              screen.blit(mapBlocks[mapData[x][y]], ((x*32)-
      (y*32)+OFFSETX,
014.                          (y*16)+(x*16)+OFFSETY -
      mapHeight[mapData[x][y]]))
015.
```

06 Building blocks

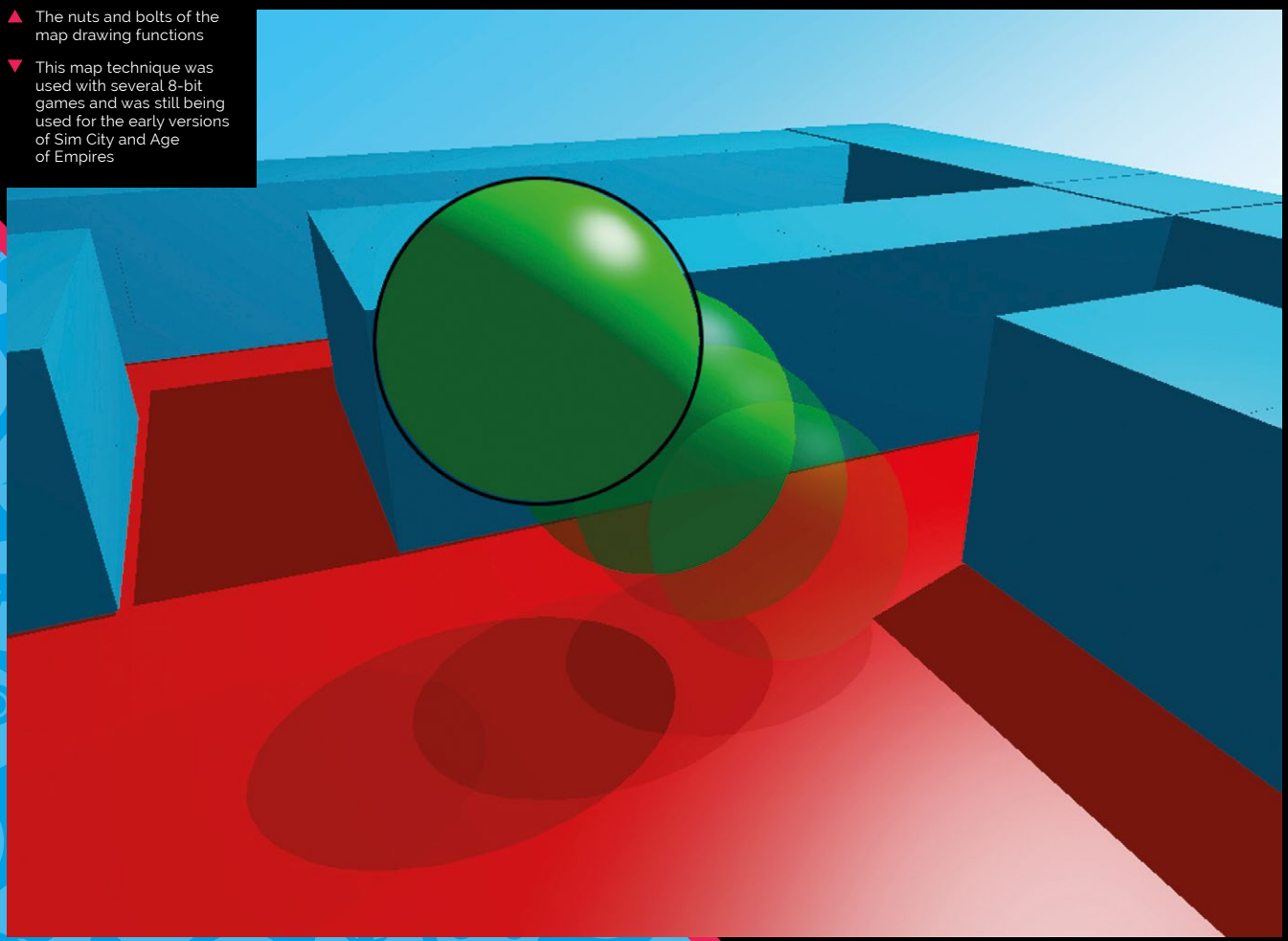
We are going to use a nested loop to run through our data and display the map. The first loop is for the x direction and inside that we have another loop for the y direction. The x and y in this case do not refer to screen pixel coordinates but to the data blocks. Now imagine that we are going to turn our data 45 degrees so that we draw our blocks diagonally down the screen. We do this with a bit of maths to translate the x and y positions in our data to locations on the screen.

07 An amazing view

To draw each block, we use the Pygame Zero `screen.blit()` function, which loads an image and draws it at the specified coordinates on the screen. So the first parameter of this function is the name

▲ The nuts and bolts of the map drawing functions

▼ This map technique was used with several 8-bit games and was still being used for the early versions of Sim City and Age of Empires



of the image. We obtain this by getting the value from `mapData[x][y]` and use `mapBlocks` to give us an image name. Now we calculate the screen coordinates. The width of each block is 64 pixels, but because we are printing diagonally we only want to move 32 pixels sideways for each block so that they overlap each other. In the down direction we move 16 pixels for each block.

08 Diamonds are forever

To get our screen x coordinate, we multiply the data x value by 32 and subtract that from the data y value times 32. That gives us a screen x coordinate starting from 0. We then add a predefined offset to move the starting block to the middle of the screen. We do the same to get the screen y coordinate but use a multiplier of 16 for the data x and y, add an offset to move the start block down the screen a bit, and take into account that some blocks are taller than others using `mapHeight`. When this runs, we'll see twelve rows and twelve columns drawn in a diamond shape.

> We'll make a dictionary to hold all the data we need to know about the player

09 Balls

Now it's time to make our player character, which in this case will be represented by a bouncing ball. We'll start by getting the ball positioned and moving around the map first. We'll make a dictionary to hold all the data we need to know about the player. See **figure3.py** for how we define the player data. The `x` and `y` values are where the player is in the block data map. We will start the player in column(`x`) 0 and row(`y`) 3. The `frame` value will tell us which frame of animation to show. The `sx` and `sy` values are the actual screen coordinates where the ball will be drawn.

10 Blitting the player

Leaving aside the other values in our player dictionary for the moment, the other piece of code in **figure3.py** goes inside our `drawMap()` loops just after we blit the blocks. The code says "if the block we are currently dealing with is the block that the player is on, work out the screen coordinates (using the same calculation as the blocks) and draw the ball to the screen." We only work out the screen coordinates of the ball once, as we'll now make a `doMove()` function that will handle the player screen coordinates from here.

11 On the move

Our `doMove()` function will introduce a couple of new Python techniques and some more of the player dictionary data. We pass `doMove()` three parameters. The first is the player dictionary structure (so we can read and write player values), and then the x and y change we want to make in block units. The `x` and `y` will be 0, 1 or -1 and represent a movement in our map block data. The first bit of `doMove()` will check to make sure that the block we are moving to is within the bounds of our map. This is a shorthand way of comparing several values and in simple terms is *0 <= x < width*, which means x needs to be between 0 and width minus 1. See **figure4.py** for the actual code to use.

Data formatting

If you are using hand-coded data, it's a good idea to format it in a way you can read it easily.

▼ Defining the player data structure and drawing the player character to the screen

figure3.py

```
001. # This code is near the top of our program
002.
003. player = {"x":0, "y":3, "frame":0, "sx":0, "sy":0,
004.           "moveX":0, "moveY":0, "queueX":0, "queueY":0,
005.           "moveDone":True, "movingNow":False,
     "animCounter":0}
006.
007.
008. # This code goes in the drawMap() function inside the y loop
009.
010.             if x == player["x"] and y == player["y"]:
011.                 if player["sx"] == 0:
012.                     player["sx"] = (x*32)-(y*32)+OFFSETX
013.                     player["sy"] = (y*16)+(x*16)+OFFSETY-32
014.                 screen.blit("ball"+str(player["frame"]),
     (player["sx"], player["sy"]))
015.
016.
```

figure4.py

```
001.  def update(): # Pygame Zero update function
002.      global player
003.      if player["moveDone"] == True:
004.          if keyboard.left:
005.              doMove(player, -1, 0)
006.          if keyboard.right:
007.              doMove(player, 1, 0)
008.          if keyboard.up:
009.              doMove(player, 0, -1)
010.          if keyboard.down:
011.              doMove(player, 0, 1)
012.      updateBall(player)
013.
014.  def doMove(p, x, y):
015.      if 0 <= (p["x"]+x) < mapInfo["width"] and 0 <=
      (p["y"]+y) < mapInfo["height"]:
016.          if mapData[p["x"]+x][p["y"]+y] == 0:
017               p.update({"queueX":x, "queueY":y,
      "moveDone":False})
018.
```

▲ Getting keyboard input and responding by setting up the data for moving the player character

12 Watch your step

After we have checked the player movement will be inside the map area, we can test to see if the movement will be to a floor block (value 0 in our data). We just need to check the value in `mapData` and we are good to go. The next line of code is a clever way of changing several values in a dictionary. We use the dictionary `p.update()` function to set `queueX`, `queueY`, and `moveDone` values all at the same time. The reason we are queueing the movement rather than just moving is because we may already be moving and we want to wait until the end of the previous move.

> "The reason we are queueing the movement rather than just moving is because we may already be moving"

Dictionaries

Organising data in dictionaries makes it easier to understand what the data is used for. It's also a good stepping stone towards using object-oriented programming (OOP).

13 The update

Staying with **figure4.py**, we can see how to get our movement controls from the keyboard. In the Pygame Zero function `update()` we look for the cursor keys being pressed and if so, call our `doMove()` function with suitable movement parameters. When we have checked for movement, we then call a function `updateBall()` which will do all the heavy lifting of animating the ball and moving it from one block to the next. You will notice that before we check the keyboard, we make sure that we are ready for more input by checking the player dictionary value `moveDone`, which we set to `False` in `doMove()`.

14 Sequencing the animation

All we have left to do now is get the ball to move from one block to the next, but if we get things in the wrong sequence we can end up with the ball being drawn in front of blocks that it is meant to be behind, or behind blocks that it should be in front of. At this stage we can bring in the changing frames of the animation to make the ball bounce up and down. We also want to make the ball move smoothly from one place to another, so the smaller the movement from one `draw()` to the next, the better.

15 You've been framed

Let's start with the animation frames for the bouncing ball. We have eight frames named **ball0.png** up to **ball7.png**. If we just increase the frame value in our player dictionary each time we call `updateBall()` and when we get to 8 set the value back to 0, our `drawMap()` function will take care of drawing the animation in a loop. The only problem with this is that if we run the animation at this speed, the ball is bouncing very fast so we need to slow it down. For this we use another value from our player dictionary, `animCounter`. With this value, we count every four frames and on the fourth frame we add one to the `frame` value.

16 Frame by frame

We need to time the movement of the ball with the correct frames so that it looks like it's

bouncing smoothly while moving. We want to wait until `frame = 4` before starting the move. At that point we move our `queueX` and `queueY` values into `moveX` and `moveY`, and set `movingNow` to `True`. When `movingNow` is `True`, the `sx` and `sy` values of the player are changed. For each block, we need to move 32 pixels left or right in 1 pixel increments, and 16 pixels up or down in 0.5 pixel increments. So our move will take 32 update cycles.

17 One block to the next

For our moving ball to be displayed correctly in the drawing order of the map, we need to change the block it is located at on the correct frame of the animation. When `frame = 7` is the time to change, so at that point we update the player dictionary `x` and `y` values based on the `moveX` and `moveY` values. We then set the player `moveDone` value to `True`. This will mean that when we return to `frame = 4`, we can clear `moveX` and `moveY`, and set `movingNow` to `False` unless another move has been queued.

18 Wrapping it up

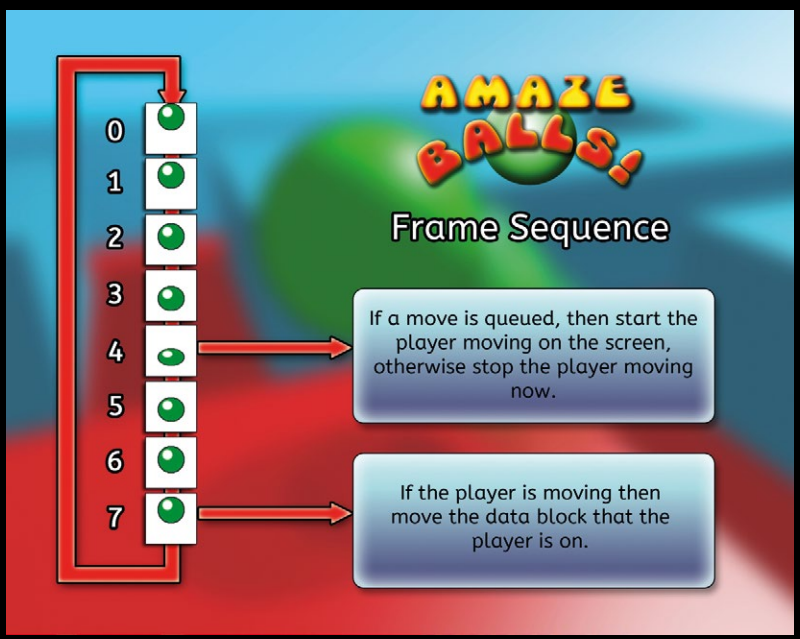
You can see from **figure5.py** how this frame sequencing works in the `updateBall()` function. You will see the check for `movingNow` first and use a separate function, `moveP()`, to change the screen coordinates of the player. Then we do all the logic around sequencing actions to the current frame. You'll also see a check to see if the maze has been solved. We set a global variable, `mazeSolved`, to `False` at the start and if the player arrives at block 11, 8 we set the variable to `True` and display a suitable message in `draw()`.

19 Level one complete

So that's the first part of this tutorial. We have looked at creating a 3D-looking map from data and 2D images, and how to move an animated character around the map. This game format has a lot of possibilities and in the next episode we'll look at making the map larger, editing the map data in an external editor, and loading the data from a separate file.

figure5.py

```
001. def updateBall(p):
002.     global mazeSolved
003.     if p["movingNow"]:
004.         if p["moveX"] == -1: moveP(p,-1,-0.5)
005.         if p["moveX"] == 1: moveP(p,1,0.5)
006.         if p["moveY"] == -1: moveP(p,1,-0.5)
007.         if p["moveY"] == 1: moveP(p,-1,0.5)
008.     p["animCounter"] += 1
009.     if p["animCounter"] == 4:
010.         p["animCounter"] = 0
011.         p["frame"] += 1
012.         if p["frame"] > 7:
013.             p["frame"] = 0
014.         if p["frame"] == 4:
015.             if p["moveDone"] == False:
016.                 if p["queueX"] != 0 or p["queueY"] !=0:
017.                     p.update({"moveX":p["queueX"],
    "moveY":p["queueY"], "queueX":0, "queueY":0,
    "movingNow": True})
018.             else:
019.                 p.update({"moveX":0, "moveY":0,
    "movingNow":False})
020.                 if p["x"] == 11 and p["y"] == 8:
021.                     mazeSolved = True
022.
023.         if p["frame"] == 7 and p["moveDone"] == False and
    p["movingNow"] == True:
024.             p["x"] += p["moveX"]
025.             p["y"] += p["moveY"]
026.             p["moveDone"] = True
```



THE Official

RASPBERRY PI PROJECTS BOOK

VOLUME 4

200 PAGES OF IDEAS & INSPIRATION

DIY Games Console

Build a Robot

Make a Magic Mirror

Set up a Spy Cam

55 PROJECTS & GUIDES

FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE

THE OFFICIAL RASPBERRY PI PROJECTS BOOK VOLUME 4

200 pages of ideas & inspiration

£12.99
200 pages of
Raspberry Pi

THE Official
RASPBERRY PI PROJECTS BOOK
VOLUME 4

Amazing hacking and making projects
from the makers of The MagPi magazine

Inside:

- How to get involved with the Pi community
- The most inspirational community projects
- Essential tutorials, guides, and ideas
- Expert reviews and buying advice

Available now magpi.cc/store

plus all good newsagents and:

WHSmith BARNES&NOBLE

Available on the App Store

GET IT ON Google Play

Build your own telephone exchange

MAKER
PJ Evans

PJ is a writer, software engineer, and Raspberry Jam wrangler. Your call is important to him.

mrpjevans.com

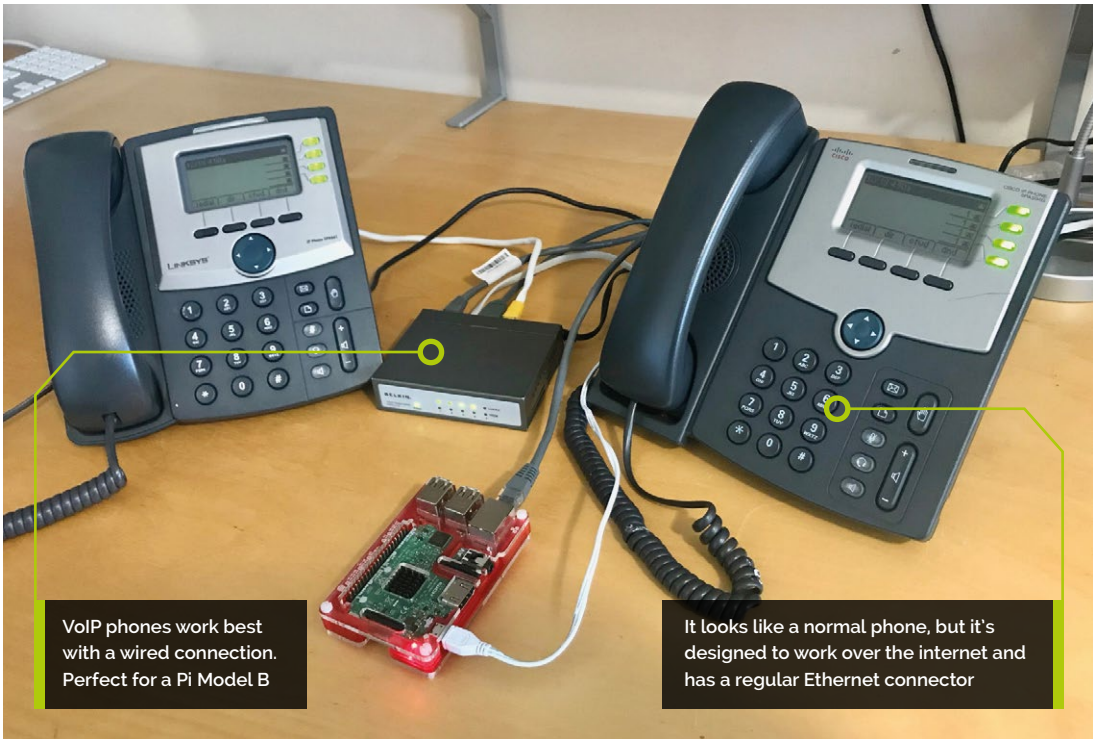
@mrpjevans

Transform your humble home phone line into a feature-packed PBX with Raspberry Pi and Asterisk

Traditional private telephone exchanges (PBX) are essential for businesses but come at a premium. Even a simple system can cost thousands to implement. Yet, with the help of Voice over Internet Protocol (VoIP) and the popular open-source telecoms platform Asterisk, the humble Raspberry Pi can match the features of the most expensive PBX systems. Small businesses can get up and running with a two-phone PBX including all the cool features for under £100. At home, why not have phones for your study or kitchen? Leave a voicemail for the dog!

01 Choose your device

One of the great things about VoIP phone systems is the flexibility. You're not constricted to a traditional phone. There are three main options. We're using the popular Linksys SPA941 and Cisco SPA504G SIP VoIP phones. These are older devices, so you can pick them up on auction sites for as little as £10. Many other types of phone are available, including DECT-style cordless. Alternatively, you can use an ATA converter that adapts a traditional PSTN phone to VoIP (although


VoIP phones work best with a wired connection. Perfect for a Pi Model B
It looks like a normal phone, but it's designed to work over the internet and has a regular Ethernet connector

You'll Need

- 2 × VoIP devices e.g. **magpi.cc/mUkceP**
- RasPBX Image **magpi.cc/tUHyFb**
- Wired Ethernet and switch
- iOS/Android device (optional)

▲ You'll need an Ethernet switch – if you run out of ports, most VoIP phones have a pass-through connector to chain other devices

you lose some features). Finally, 'soft' phones such as Zoiper can run on iOS/Android devices or on your computer.

02 Network setup

VoIP systems rely on fast networks with low latency to maintain audio quality. As a result, many VoIP devices, including our choice of phones for this tutorial, do not feature WiFi. For best performance, we're going to wire everything up 'the old-fashioned way' using a small Ethernet switch. Most VoIP phones feature a passthrough Ethernet connector, so you can chain other devices without using up all your switch's ports. The phone can be set to take priority over the data if required (QoS or quality of service) so phone calls still work if you're downloading the latest Fortnite update.

03 Prepare the phones

If you've procured your phones second-hand as we did, they'll probably need resetting before going any further. VoIP is not one protocol but many working together and the range of possible settings runs into the thousands. To avoid frustration later on, it's vital to ensure that each phone has been fully reset to defaults and is successfully talking to the network. On this particular phone range, it can be done by going into Setup (the 'paper' key) and then 'Factory Reset'. Allow the phone to restart and then check Setup > Network. All being well, the phone's IP address will be displayed.

> "To avoid frustration later on, it's vital to ensure that each phone has been fully reset to defaults"

04 Install RasPBX

Asterisk is a popular and mature software implementation of a traditional PBX. The capabilities and features of this product are staggering, especially given its open-source status. To paraphrase: with great power comes great configuration. Asterisk can be hard work to set up. Luckily, some dedicated folks have put together RasPBX, a Stretch distribution with Asterisk installed and including FreePBX, a

Top Tip

Performance matters

VoIP audio codecs rely on a good network. Consider a fast switch for best performance.

▲ VoIP phones such as this Linksys SPA941 are easy to find second-hand and are full of features

web-based front-end. Download the image from **raspberry-asterisk.org** and burn to a microSD card (we used Etcher). Installation requires a few steps and these are detailed at: **magpi.cc/pgmfnY**.

05 Set up FreePBX

Once you have the basic configuration in place, open up a web browser then open **http://raspbx.local/**. If this doesn't work, establish the IP address of your Pi using a tool like Fing for iOS/Android, as we'll need it later. FreePBX will now start up and walk you through a simple process where you'll create an administrator account for the system. When complete, the number of options and screens may be a bit bewildering, but don't worry: RasPBX is very sensibly set up by default and we don't need to change anything to get up and running.

06 Create extensions

In VoIP/PBX terminology, each endpoint for a call is known as an extension rather than a phone, as most VoIP phones can handle multiple extensions if desired. Let's keep it simple with one per phone. From the FreePBX homepage, click on FreePBX Administration, log in as the administrator, then click Applications followed by Extensions. On the page, click Quick Create Extension. Choose a unique extension number (we imaginatively went with '1'), and a display name (the user). Repeat for all the extensions you want, remembering to click Apply Config for settings to be reloaded.

07 Configure the phones

This will vary depending on which phone or device you have. What's important to know: the IP address of the Pi, the extension number, and the 'secret' (password). To get the secret, click the Edit icon against each of the extensions you have created. Now get the IP address of the phone you are using. Chances are there is a web interface. On our phones' websites, we clicked Admin Login, then Ext 1. We then added the Pi's IP in Proxy, the extension in User ID, and the secret in Password. After a reboot, and all being well, your phones should register with RasPBX.

08 Make a call!

In the case of the Linksys/Cisco range, a successful 'registration' results in a display of green lights indicating the phone is ready to take calls. Let's say you have extensions 1 and 2. Pick up the receiver of extension 1, press 2 on the keypad and then hit Dial. If everything is running as it should, your other phone will ring. Get someone to answer it (or talk to yourself, we won't judge) and check audio is flowing correctly. You have a PBX! Try leaving one phone off the hook and calling it: you'll get a voicemail response ready to go.

09 Voicemail

One of the most obvious features we need is some kind of voicemail facility. RasPBX has you covered. Without any further configuration required, you have a full voicemail system up and running. Access is via a 'feature code', a number with special meaning normally prefixed with '*'. For voicemail it is *97. In our phone's menus we were able to map it to the voicemail button. The first time you call, you'll be guided through setup so you can allocate a PIN and personalised greeting messages. With a bit of advanced tinkering, you can also have the messages emailed to you as attachments.

10 Features, features, features

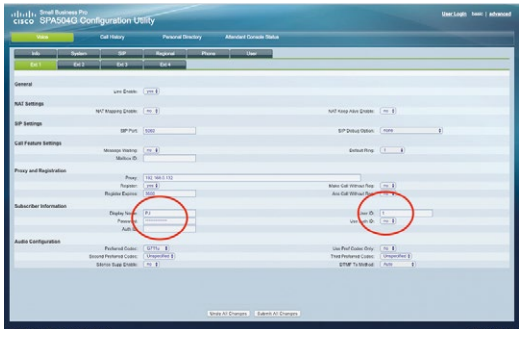
Now you have the basics up and running, there is plenty to play with. Standard PBX functions such as call transfer, call groups (conferencing), and do-not-disturb are configured to work right out of the box. Have a look around the FreePBX interface, especially the Applications section, and you'll find additional powerful features. Try call recording, dedicated conference facilities, ring groups (where an incoming call can 'move' from extension to extension, or ring several extensions simultaneously), and even IVR ("press one for, press two for"). If that's not enough, there are even more advanced features available as commercial add-ons.

11 Soft phones

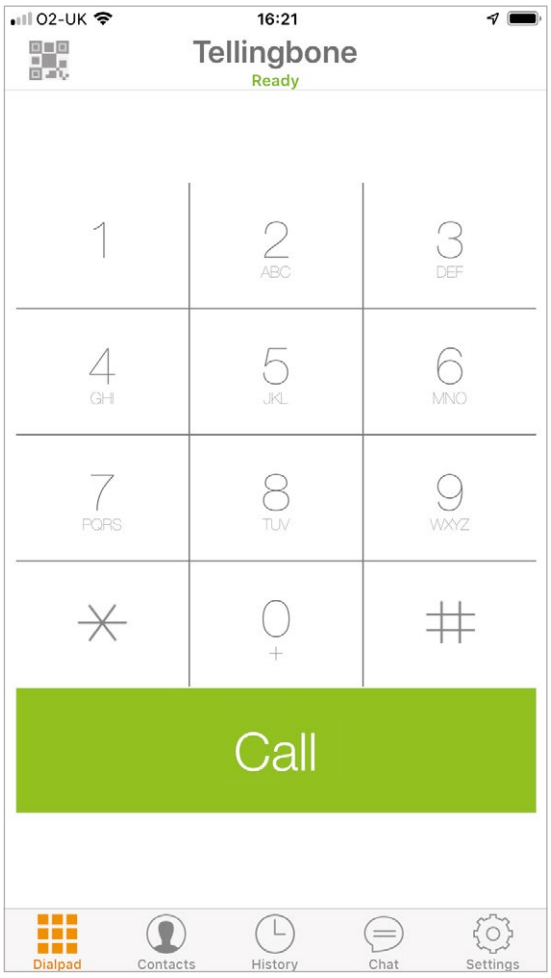
As we mentioned at the start of this tutorial, VoIP phones are not your only option. There are some good 'soft' phones available. As the name suggests, these are software-only and designed to run on your smartphone, tablet, or computer. A popular choice is Zoiper, which is available for Windows, macOS, Linux, Android, and iOS. Once installed, add a new account and choose manual setup. Enter your extension number as the user name, your secret as the password, and your Pi's IP address as the 'domain'. Click on Register and within a few seconds the phone will be online.

12 Trunking

To send and receive external calls, we need a provider that knows how to take our request and route it on to the public network as well as handle incoming calls. This is known as a SIP trunk and there are many available with varying costs. A popular option is Voipfone (**voipfone.co.uk**), which offers a free trunk service (excluding call charges). A lot of configuration is required to create an inbound and outbound route, but Voipfone has provided detailed instructions to get you started. Your PBX can now speak to anyone, anywhere in the world with a phone.



▲ Configuring a VoIP phone: just change User ID and Password



◀ Your Raspberry Pi PBX will also work with 'soft' phones such as Zoiper, which runs on many different devices

13 Extend and expand

Even though our little Raspberry Pi can handle many extensions and all these features, there is yet more it can do. Some advanced add-ons for Asterisk are well supported by RasPBX. You can add a call-barring list to avoid scammers and cold-calls, add in a fax-to-email gateway should you need it, and even use a 3/4G GSM USB dongle to provide call routing over the mobile phone network – perfect as a failover measure. There really is very little an expensive PBX can do that RasPBX can't for a fraction of the price.

Top Tip

Made no difference?

If you've made changes in FreePBX and they haven't taken effect, look out for Apply Config in the top-right corner.

Part 02

Pi Bakery: In a spin

Enhance your Raspberry Pi by adding a VGA output and a retro 8-bit sound chip emulator using the eight-core Propeller chip. This month, exploring sound

MAKER

Mike Cook

Veteran magazine author from the old days, writer of the Body Build series, plus co-author of *Raspberry Pi for Dummies*, *Raspberry Pi Projects*, and *Raspberry Pi Projects for Dummies*.

magpi.cc/TPaUfT

Last month we saw how to put together the hardware for a Propeller development system we could use with the Pi. Now we see how this board can be programmed using the Propeller IDE (integrated development environment) and how we can communicate with it using both the IDE's terminal and Python. Then we will see how to emulate the classic SID sound chip from the Commodore range of computers, and control it from Python. Finally, we will see how to use this chip over MIDI as we turn it into a three-note polyphonic synthesizer.

01 Installing the IDE software

First off, we have to install the IDE (integrated development environment) for the Propeller processor. Open a Terminal and enter:

```
wget https://github.com/parallaxinc/PropellerIDE/releases/download/0.36.7/propelleride-0.36.7-armhf.deb
```

```
sudo apt-get install libqt5serialport5
sudo dpkg -i propelleride-0.36.7-armhf.deb
```

You will then find an entry for the Propeller IDE under the Programming submenu of the Raspbian applications menu. Select this to get the screen shown in **Figure 1**. To finish the install, select one of the items in the help menu and you will be asked for an application to read PDF files. Select the PDF reader in the Utilities submenu. Under the Edit > Preferences menu of the IDE, choose the 'theme' tab and select one you like – see **Figure 2**.

02 Your first program

The first thing you do in almost any new physical computing project is to get an LED to flash. We have an LED on pin 26 of our development board and we are going to blink this. So click on the New File icon – the leftmost one on the icon bar at the top of the IDE. Then type in the code shown in the **blink.spin** listing. Next,

You'll Need

- Propeller development board from *The MagPi* #77 **magpi.cc/77**
- Active (powered) speaker
- MIDI keyboard

Figure 1

▶ **Figure 1**
Our preferred colour theme for the IDE

> This code is written in the Spin language, and might look a little odd at first

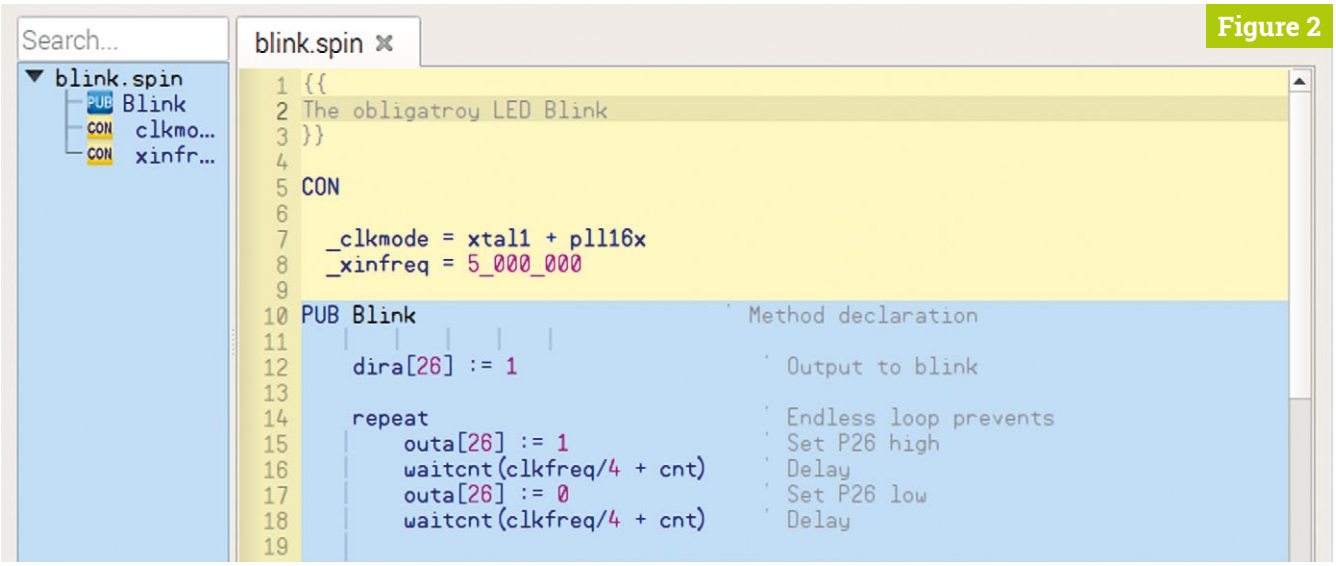
click on the hammer icon to check it will build correctly and select the serial interface to use from the drop-down menu to be 'ttyUSB0'. Now, click the right-pointing arrow to download and run the code.

03 Understanding the syntax

This code is written in the Spin language, and might look a little odd at first, but is not too different from any other procedural language such as Python. A Spin program's structure first defines any constants, then calls objects which are programs for other cogs (processors). Then the functions are defined and finally, data lines. The use of the equals sign is reserved for defining constants – assigning values to variables uses the := operator, which can be thought of as 'becomes equal to'. We recommend you put all cog files for one project in the same folder.

▼ An alternative active speaker

Figure 2



▲ **Figure 2** One of the alternate colour themes for the IDE

04 Using resources

What makes the system powerful is the use of resources or cogs that others have written. An excellent source is to be found at the Parallax Object Exchange site at **magpi.cc/JqaPdV**. These save you from reinventing the wheel, which makes things a lot simpler. Also, many simple examples can be found by downloading the PDF file found at **magpi.cc/sZBzVx**. The best way to get a feel for programming with Spin is to play about with the code and change things. If you want a more formal introduction to Spin, we would recommend the book *Programming the Propeller with Spin* by Harprit Singh Sandhu.

05 Communication

The **SerialAdd.spin** program demonstrates the simple use of communication between the Spin processor and the Pi's keyboard. It inputs two numbers, adds them up, and displays the result. This code calls up a cog called **FullDuplexSerialPlus.spin**, which can be found in our GitHub repository or from the Object Exchange, and should be placed in the same folder as **SerialAdd.spin**. After checking it will build using the hammer icon, this time you should run the code by clicking on the down-pointing arrow; this will load the program into EEPROM so it will be run automatically on a system reset and power cycle.

06 Using the IDE's terminal

Open the IDE's terminal, select 'ttyUSB0', and click on the Reset tab. Click to remove the tick from the Echo box and then click in the terminal's window to give it the cursor focus. When asked for two numbers, type them in (followed by **RETURN**) to display the sum of those two numbers. This will then repeat. Note that you will not see what you type in immediately, but only when you have pressed **RETURN**. We can use this cog to send and receive data from Python using the serial module in a very similar way.

07 Exploring sounds

Go to the Object Exchange site and select the Speech/Sound section and try some of the examples. We would recommend you try the singing demos from Chip Gracey and also 'Joy to the world'. Plug an active speaker (amplifier and loudspeaker) into the jack socket on the board. Also worth a look is the Propeller Signal Generator, especially if you have an oscilloscope to view the waveforms. Finally, download the SIDcog v1.3 and run the Example play routine code. Note that the Dump Play won't work because our board does not have an SD card reader on it.

> "The program will be run automatically on a system reset and power cycle"

08 The SID chip

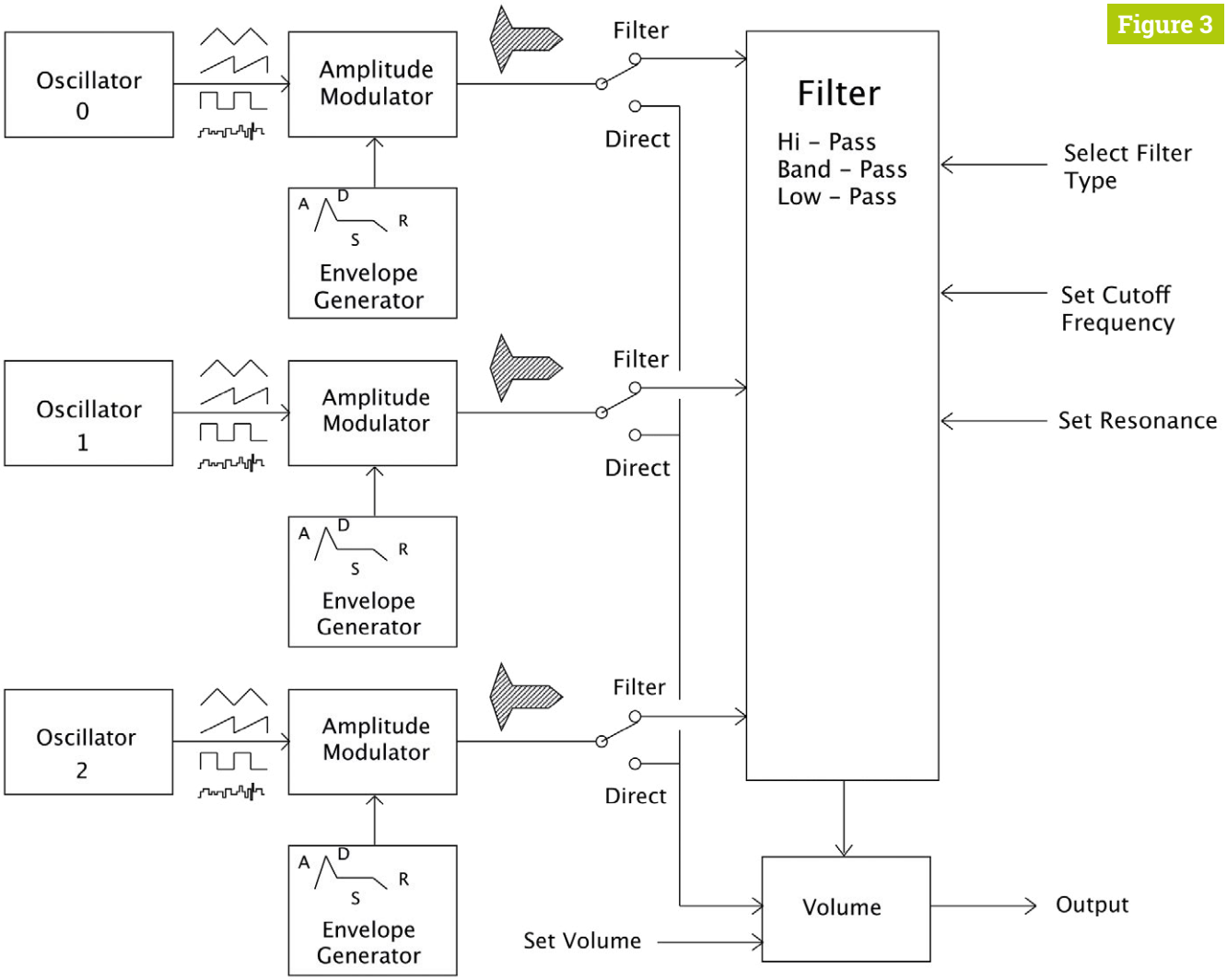
The MOS 6581 SID (Sound Interface Device) chip is the most capable of the sound generator

Top Tip

Default serial port

The IDE uses a serial port. It defaults to 'ttyAMA0', which used to be the GPIO's serial port. However, following the introduction of Bluetooth on the Pi 3 models, this has changed to 'ttyS0', which is not picked up on the IDE, so you have to select 'ttyUSB0'.

Figure 3



▲ **Figure 3** Block diagram of the SID chip

chips fitted into early computers. It consists of three waveform generators, each followed by an ADSR (attack, decay, sustain, release) envelope-shaping circuit. It then has an optional filter that can operate on any or all of these waveforms. It's also capable of having the filter parameters controlled by external components, which makes exact emulation difficult. The block diagram is shown in **Figure 3** and it is worth downloading a data sheet to see the register structure of the chip.

09 Talking to SID

Songs are available in the .sid file format, but these are complex, as they consist of not only the song data, but also the machine code required to put the right data in the right place at the right time. So they are a bit tricky to emulate. We won't try to do this here because there are a few .sid file players about and we want to do something new.

On our GitHub we show you examples of sending data to a SID cog from Python, but what we want to do is to play the SIDcog as if it were a MIDI sound generator.

10 Polyphonic

There are two approaches to making a SID chip playable from MIDI: monophonic or polyphonic. That is, only one note at a time or multiple notes at the same time. There are trade-offs with each approach; with monophonic, you can make more different sorts of sounds but you can't play chords. We decided to go down the polyphonic route for a much more playable instrument. With three waveform generators in the chip, we decided to make an instrument with a polyphony of three. So the waveform, filter switching, and ADSR envelope of each oscillator are the same.

blink.spin

DOWNLOAD THE FULL CODE:
magpi.cc/XCwzVh

> Language: **Spin**

```
001.  {{
002.  The obligatory LED Blink
003.  }}
004.
005.  CON
006.
007.    _clkmode = xtal1 + pll16x
008.    _xinfreq = 5_000_000
009.
010.  PUB Blink                          ' Method declaration
011.
012.      dira[26] := 1                   ' Output to blink
013.
014.      repeat                          ' Endless loop prevents
015.          outa[26] := 1               ' Set P26 high
016.          waitcnt(clkfreq/4 + cnt)    ' Delay
017.          outa[26] := 0               ' Set P26 low
018.          waitcnt(clkfreq/4 + cnt)    ' Delay
```

SerialAdd.spin

```
001.  ''----------SerialAdd.Spin-----------------------
002.  CON
003.    _clkmode = xtal1 + pll16x ' 80MHz internal clock
004.    _xinfreq = 5_000_000
005.
006.  RxPin = 31     'default boot loader RX
007.  TxPin = 30     'default boot loader TX
008.  BaudRate = 115200 'this is a good baud rate for 80MHz
009.  OBJ
010.    SerIO : "FullDuplexSerialPlus" 'object that implements serial I/O for the Propeller.
011.
012.  PUB main | N1, N2' The prime function in Cog 0 and first run on boot
013.  SerIO.Start(RxPin, TxPin, %0000, BaudRate) ' initialise SerIO object
014.  SerIO.Str(String("Adding up two numbers ")) ' Introduction text
015.  SerIO.tx(10)                        ' send LF                                    '
016.  repeat
017.     SerIO.tx(10)                     ' send LF
018.     SerIO.Str(String("Input two numbers "))
019.     N1 := SerIO.GetDec               'get dec number i.e. text numeric
020.     SerIO.Dec(N1)
021.     N2 := SerIO.GetDec
022.     SerIO.Str(String(" + "))
023.     SerIO.Dec(N2)                    'get next one
024.     SerIO.Str(String(" = "))
025.     N1 := N1+N2                      'add them
026.     SerIO.Dec(N1)                    'send back the result as text
027.
```

Top Tip

Errors in online demos

Many examples found online have been written on a case-insensitive filing system. As a result, you might get a 'file not found message' when you try to compile them (hammer icon). If you do, simply change the offending file's name to match the case of the file given in the error message.

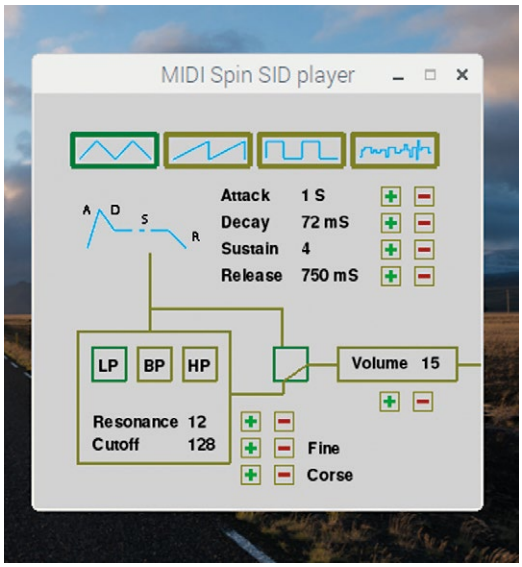


Figure 4 ▲ **Figure 4** The graphical user interface to the MIDI player

11 Polyphonic Pi

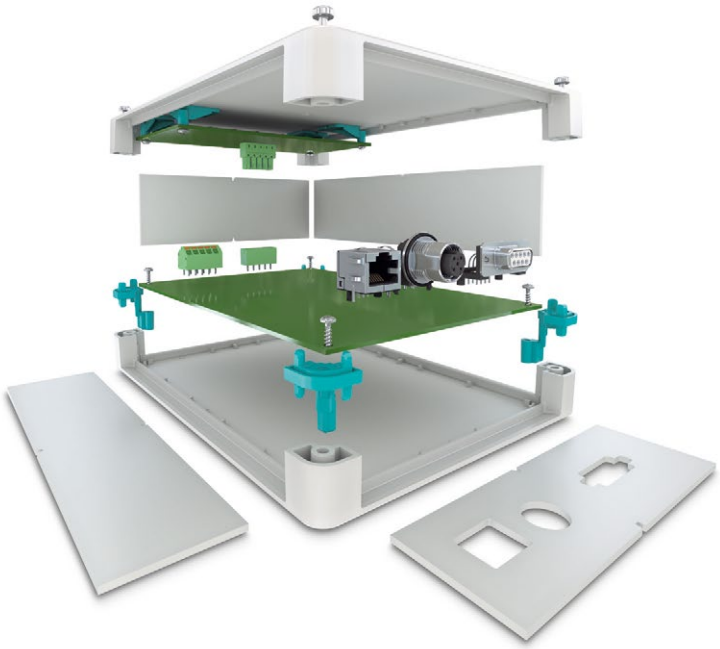
We chose the trusty Pygame framework to handle the MIDI input and, more importantly, the polyphony. This is quite tricky as not only have we to sort out what SID channel to play an incoming note on, but also, if there is no free channel, we must release a channel by turning an already playing note off before we play the new one. This is further complicated by the fact that we don't want to turn off just any note, but the note that has been playing the longest. In addition, we wanted a graphical interface to the SID chip's control parameters.

12 MIDI keyboard player

As the MIDI player code is long, we don't have the space to print it here, but it is available on our GitHub page. **Figure 4** shows the graphical interface; each parameter can be controlled by a mouse click. The ADSR parameters are not continuously variable but are fixed unequal size steps; the + and – boxes step through these. The filter block can be switched in or out and the sort of filter changed. Note you must have at least one sort of filter selected to get any sound out. If your MIDI keyboard has controls built in, these can be programmed to change the parameters as well.

Next time...

There is lots more to explore with the SID chip and sound from the Propeller board, but in next month's final part we will look into generating a VGA output from our board.



The versatile home for single board computers

Multiple mounting option enclosure system

The new UCS Universal Case System can be wall, desk or DIN rail mounted giving complete flexibility for your design. Suitable for embedded systems & single board computers the enclosure is available in black or grey, in 4 sizes, 2 heights and Raspberry Pi 3B & 3B+ pre-milled versions.

For additional information call 0845 881 2222 or visit **phoenixcontact.co.uk/UCS**

PHOENIX CONTACT
INSPIRING INNOVATIONS

Easy compilation

Here we look at how to install software and build it from the source code

AUTHOR

Richard Smedley

Having often found words better than pointing at things, Richard stuck with the command line whence all around had fled.

@RichardSmedley

It's so easy to install most software on Raspbian – provided it's mature enough to have been packaged as a DEB (.deb) archive. Often, however, there's some great code available you'd love to try, but it asks you to compile it – sometimes after first cloning it from GitHub.

While not so straightforward as running an `apt-get install` command, there's little to fear in stepping through the decades-old ritual of compiling software – and when errors do occur, it's often quite easy to get back on track. We'll also look at scripted installs and Python packages, but first let's find out what to do if a package is in the wrong format.

Raspbian isn't the only distribution of GNU/Linux based upon Debian – **distrowatch.com** lists more than a hundred – and the ones you're most likely to have heard of include Linux Mint and Ubuntu. Ubuntu – the name derived from an Nguni Bantu term meaning 'humanity towards others' – is so popular that many projects, including ones not in the Debian and Raspbian repositories, maintain DEB packages for each version released.

Ubuntu also introduced the idea of Personal Package Archives (PPA) to the Debian family – special software repositories for uploading source packages to be built and published as an APT repository by Ubuntu's Launchpad software like the official Ubuntu repository, but for unofficial software from outside. You won't tend to find these with Pi software, but if you've been inspired to try one of Raspbian's relatives on your main PC, you'll find plenty of instructions on them in the Ubuntu community documentation.

Track your installs

Installing software from outside of Raspbian's repository means it must be looked after separately: updates, bug-fixes, security patches, and disaster recovery if something happens to your Pi. Keep track of what you install.

Available packages

When you run `apt-get`, or you `apt-cache search` to look for a package, APT quizzes its local record of what packages are available to it. The record of where it gets these packages from – the address of the repositories – is kept in the file **/etc/apt/sources.list** and at files in **/etc/apt/sources.list.d/** to which you can also add repositories by hand should a project you are interested in maintain one.

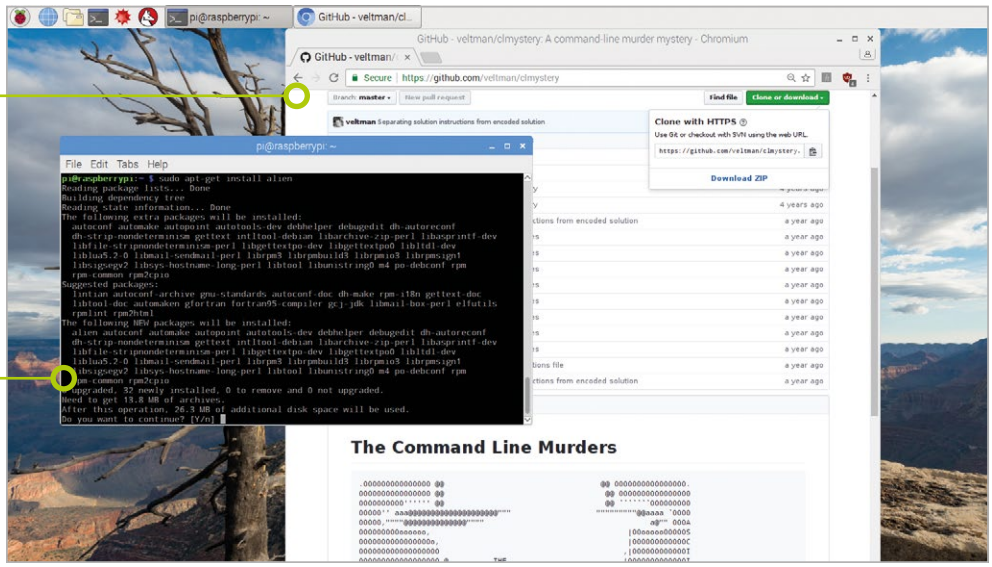
You can also edit **sources.list** (not advised) – editing all mentions of wheezy to jessie was an (unsupported) way of upgrading without overwriting your SD card when Raspbian updated. Meanwhile, plain old DEB files can be downloaded, and then installed with the command:

```
sudo dpkg -i example.deb
```

And any missing dependencies resolved with a:

```
sudo apt-get install -f
```

There is another popular family of GNU/Linux distributions, based upon Red Hat, and including Fedora and CentOS. Fedora is available as an alternative to Raspbian (on the Pi 2 or 3), should you wish to try it out. What we're concerned with


Just paste the URI ending .git after git clone at the Terminal and you'll have the very latest source code
Install alien and you'll be able to swap packages between Raspbian, Fedora, and other systems' packages
The Command Line Murders

here is situations where you may need to install packages for one distribution, onto a system of the other type. Your friend here is Raspbian's alien package, which will convert software between DEBs and packages in RPM (Red Hat Package Management) format.

```
alien some-package.rpm
```

...will convert from RPM to DEB. You're more likely – on the Pi at least, where RPM-only packages are rarer – to need to convert the other way:

```
alien -r mysoftware.deb
```

Friendly triad

Before package managers, it was normal to compile your own software, obtaining source code – usually written in the C or C++ languages – and running it through the GCC compiler, before linking libraries, and installing to the correct place on your disk drive.

Many of the headaches involved in the process are long-gone as `configure` and `make` scripts do all of the hard work, checking dependencies are installed, then running the correct compiler flags for the project and the platform, and even installing the man page in the correct place.

Although most of the software you'll want to run will be available as a DEB to install with `apt` or `dpkg`, or come with a shell script which deals with installation (see below), plenty of projects, particularly those you'll find on GitHub or FreshCode, need unpacking then compiling.

After unpacking the archive with:

```
tar xvf latest-software.tgz
```

...`cd` into the directory created, and look for a file called **README** or **README.md**, or perhaps one called **INSTALL**. It will usually tell you to run a trio of commands that will become familiar – but read the instructions as there are variations, and some software even bypasses make with its own local version which you'll need to run as `./make`. The norm is:

```
./configure
make
sudo make install
```

However, on the Pi, `./configure` on its own may well result in programs (and particularly libraries) being built so they install in a not-quite-standard location – **/usr/local/lib** instead of **/usr/lib**. This can cause problems if these directories aren't on the library search path. In particular, if building a new version of a library that is already on the system, you will often end up with two versions of the library on the system – and the system will continue to use the old one. So, instead of plain `./configure`, we advise using the following to avoid issues:

```
./configure --prefix=/usr --libdir=/usr/
lib/arm-linux-gnueabihf
```

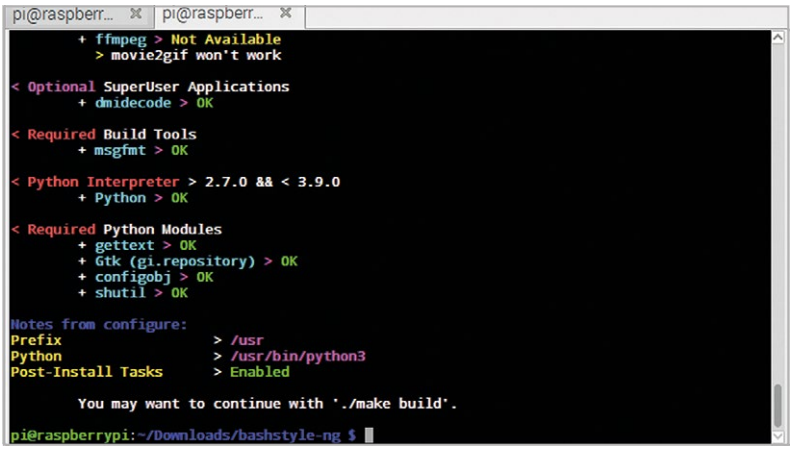
Dependencies

Commercial software (open-source or proprietary) often comes as a large binary file

Top Tip

Git book reviews

We've reviewed a couple of useful books on Git in *The MagPi* – take a look at the reviews in issues 41 and 58.



▲ Compilation can look complicated, but most projects provide a well-crafted make file for the process – a custom one even providing hints here

containing all of the dependencies, statically linked into the application. With most non-commercial software projects, however, it is more common to just get a list of which version of which library will be needed to compile and run the software. Luckily, for a standard Debian package, there's a simple command that will install all the dependencies required to build it:

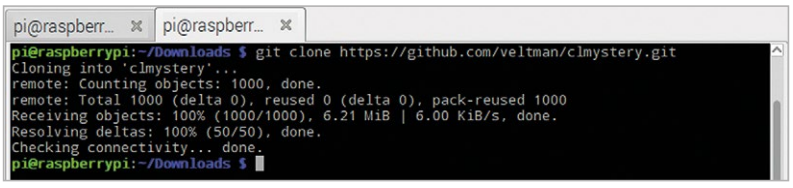
```
sudo apt-get build-dep <package name>
```

However, occasionally you'll need to compile and install something else first. Often the reason for compiling is to get the very latest version of the software from the developers, for new features, compatibility, or bug fixes. In the last few years, GitHub has become the default place to host free software projects; other repositories are available, but we'll just look at fetching software from GitHub, to quickly show you what you need to know.

> GitHub has become the default place to host free software projects

Your first encounter with GitHub's existence may be seeing a banner on a project page inviting you to 'fork me on GitHub', or an invitation to 'clone' the software. Yes, we're in the world of a project big enough to sustain its own jargon. A fork is simply your own development copy to work on, after which you can offer the changes back to the project, or publish them (on GitHub or elsewhere) for others to try or build upon.

▼ Git was developed by Linux creator Linus Torvalds to handle the complexities of multi-million lines of kernel code – yet makes it simple to maintain version control on the simplest of software


pi@raspberr...
pi@raspberr...
pi@raspberrypi:~/Downloads $ git clone https://github.com/veltman/clmystery.git
Cloning into 'clmystery'...
remote: Counting objects: 1000, done.
remote: Total 1000 (delta 0), reused 0 (delta 0), pack-reused 1000
Receiving objects: 100% (1000/1000), 6.21 MiB | 6.00 KiB/s, done.
Resolving deltas: 100% (50/50), done.
Checking connectivity... done.
pi@raspberrypi:~/Downloads $

It's a mystery

The clmystery in the GitHub example is The Command Line Mystery – a text game which teaches you command line use. Entertaining and very useful.

You don't even need to be able to code – many people are now using GitHub for collaborative development of documentation, including scientific research, and even fiction. But to simply download something, we don't need to worry about other Git methods – just 'clone' the application's source to your Pi with the following command:

```
git clone https://github.com/veltman/
clmystery.git
```

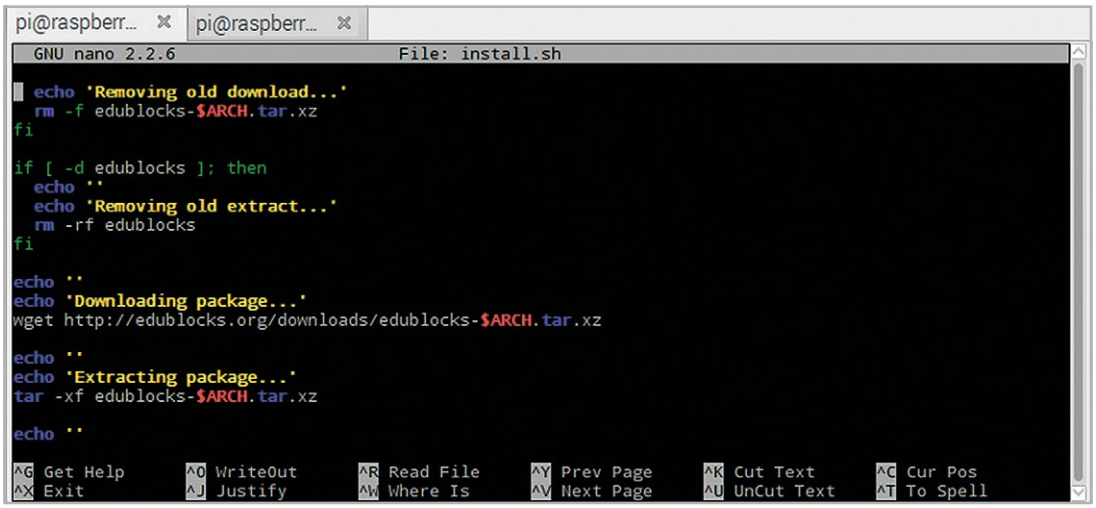
…which will make a local directory containing all of the source files. `cd` into the subdirectory just created. Then follow the configure/make instructions as above.

Nil desperandum

Sometimes, somewhere along the compilation, something goes wrong, and the script terminates with a complaint of some missing package. On a good day, you'll `apt-cache search` for the name of the missing dependency and there it will be, easy to install on Raspbian.

On a slightly-less-good day, you're going to have to dig around a bit to find the software, or the latest version that's being called for. Maybe having to go to GitHub or **SourceForge.net**. This is OK if the developer has a setup not too far removed from your own; the extra install steps may go smoothly. Otherwise, don't despair: help is available.

Most projects have one or more ways of reporting a problem and seeking help: a mailing list or Google Group; a wiki on the project SourceForge page; an email address for the lead developer; or even a Twitter account. Try to state your problem with plenty of detail, and remain polite and patient, and usually you'll find helpful people. Remember, although people want to be helpful, you are asking them to give up time to answer you – if you're frustrated with the installation process, and we've all been there, don't let that stop you being respectful of anyone going out of their way to help you.


pi@raspberr...
pi@raspberr...
GNU nano 2.2.6 File: install.sh
echo 'Removing old download...'
rm -f edublocks-$ARCH.tar.xz
fi
if [-d edublocks]; then
echo ''
echo 'Removing old extract...'
rm -rf edublocks
fi
echo ''
echo 'Downloading package...'
wget http://edublocks.org/downloads/edublocks-$ARCH.tar.xz
echo ''
echo 'Extracting package...'
tar -xf edublocks-$ARCH.tar.xz
echo ''
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

EduBlocks is a good example of a project that makes your life easy by putting every stage of the installation into a shell script, and giving you just one command to run the installation

If you don't have time for chasing up answers, but remain interested in a project, another option is to wait for the project's next release. By this time the dependencies may have become more widely available, even making their way into the Raspbian repositories, or the problem may simply no longer arise.

A scripted install

You'll sometimes be asked to grab an installation script and run it directly – as is the case with the excellent EduBlocks, a half-way house between Scratch and Python which is helping young coders get over the large step between the two languages. The EduBlocks install wants you to run:

```
curl -sSL get.edublocks.org | bash
```

This takes the shell script at **get.edublocks.org** (you can look at it there in your browser) and pipes it through to a Bash process to run. The `-s` switch tells curl not to show a progress meter or error messages; `-S` overrides this to show an error message if it fails; while `-L` tells it to follow to wherever the site redirects to for the script.

If you don't like running software without knowing what it is doing to your Pi, or simply wish to take a look inside the script and see what the installation does, instead, download and save as **edu-install.sh** or just **install.sh** if that won't overwrite anything of the same name in your current working directory:

```
curl -o install.sh -L get.edublocks.org
```

...and you'll see that the script downloads and unpacks a tarball with more than one script inside, first running the one that installs dependencies.

You can run the downloaded script – which you can do with `sh edu-install.sh`. If you don't read the script before running it, you're placing a lot of trust not just in the developer(s) who wrote it, but every step of its journey before it reached your machine.

Python, please

Over the years of installing software from various sources, you learn to recognise signs of whether or not it's likely to be a painless installation – and one cause for hope is something written in Python. Although both good and bad software can be written in any language, Python software and Python libraries just seem to be well-packaged and reliable.

Although Debian-based GNU/Linux distributions like Raspbian come with the excellent APT and dpkg (Debian Package Manager), many popular programming languages have evolved their own ecosystems of packing tools and repositories – there are several for the Emacs text editor alone! Two that you are most likely to come across are JavaScript's npm and Python's pip. After using apt, the command format will be familiar, and:

```
sudo pip3 install numpy
```

...won't be a stretch to remember if the installation instructions tell you to install Python libraries such as NumPy. Happy installing, and above all, don't worry – it's quite hard to mess up the Raspbian installation, but if you do, at least those backups you made in last issue's tutorial (**magpi.cc/77**) will get tested out.

Going for Ubuntu

Ubuntu, can be installed on the Pi (2 or 3), as well as your PC, Mac, or laptop,

Make a Sense HAT MP3 player

Turn a Raspberry Pi and Sense HAT accessory into an MP3 player

MAKER

Marc Scott

Marc is a former teacher of Computer Science, and is currently a learning manager at the Raspberry Pi Foundation.

magpi.cc/ZHdSsd

In this project, you will be making your very own MP3 player using a Raspberry Pi and a Sense HAT. You'll be able to shift between tracks in your playlist, change the volume, and show a cool disco display on the LED matrix.

You'll Need

- Raspberry Pi (not a Zero / Zero W)
- Sense HAT
- Speakers or headphones

01 Get set up

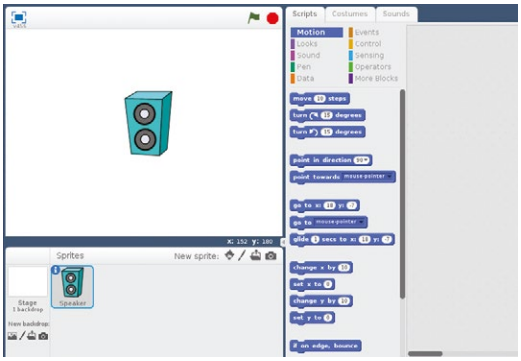
Open up Scratch 2 on the Raspberry Pi (choose Menu > Programming > Scratch 2). Now choose a sprite for your MP3 player. Right-click on the cat sprite and choose Delete. Click on the 'Choose sprite from library' icon; now choose Things in the sidebar and highlight Speaker. Click OK to add it to the Stage. It will add some blocks to the scripts area – right-click to remove them.

02 Add GPIO support

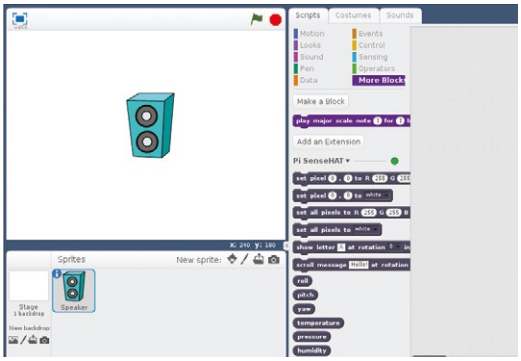
The Pi GPIO extension in Scratch 2 on the Raspberry Pi is needed to control and read input and output components connected to the GPIO pins. With Scratch 2 open, click on More Blocks in the Scripts menu. Next, click on the 'Add an Extension' button. You can then double-click on the 'Pi SenseHAT' extension to add it. You should now see the Pi SenseHAT blocks in the More Blocks section.

03 Create a disco effect

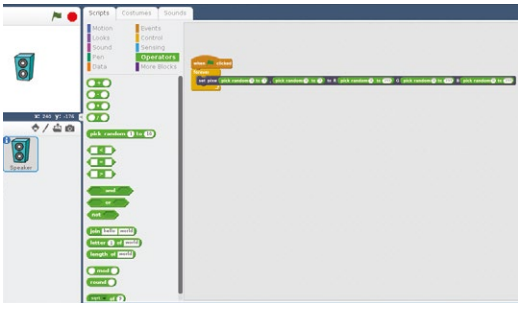
Click Events and drag a `when flag clicked` block to the scripts area. Choose the Control tab and add a `forever` block below `when flag clicked`. Within the More Blocks menu, find the `set pixel` block and drag it inside the `forever` block.



▲ **Step 1** Add the Speaker sprite

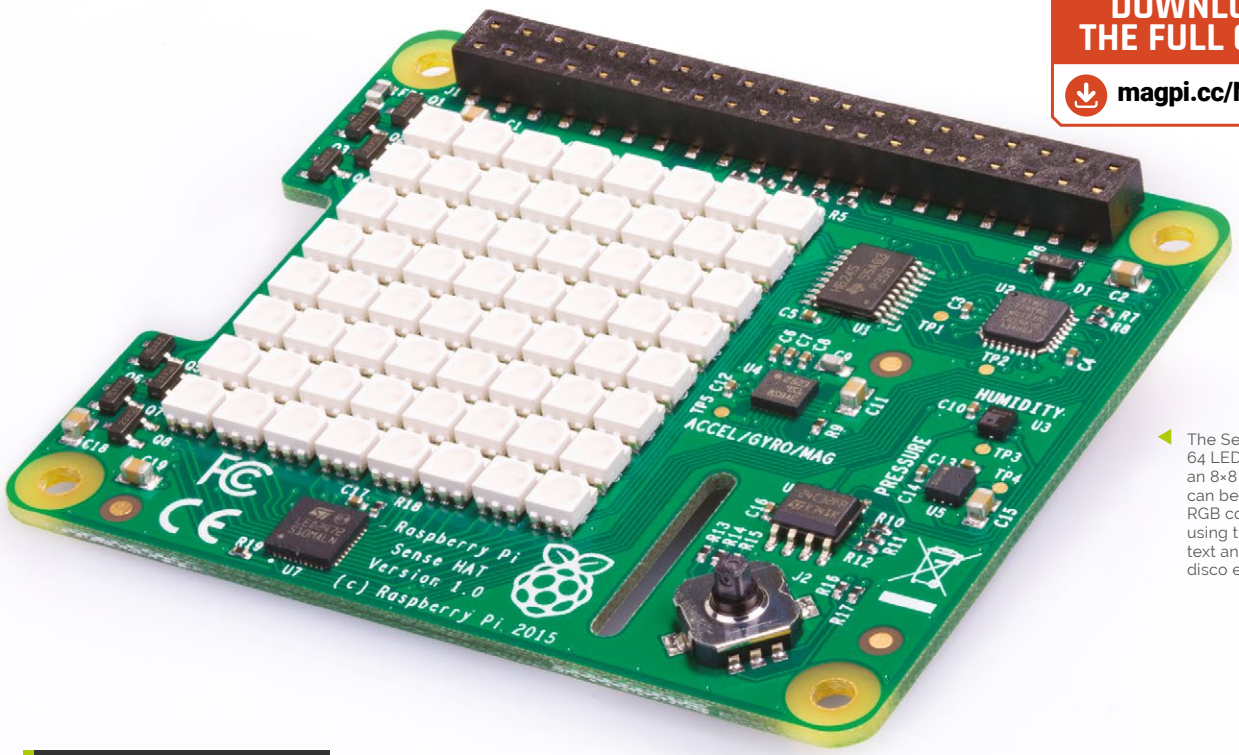


▲ **Step 2** Pi SenseHAT blocks appear under More Blocks



▲ **Step 3** Creating a random sparkle effect

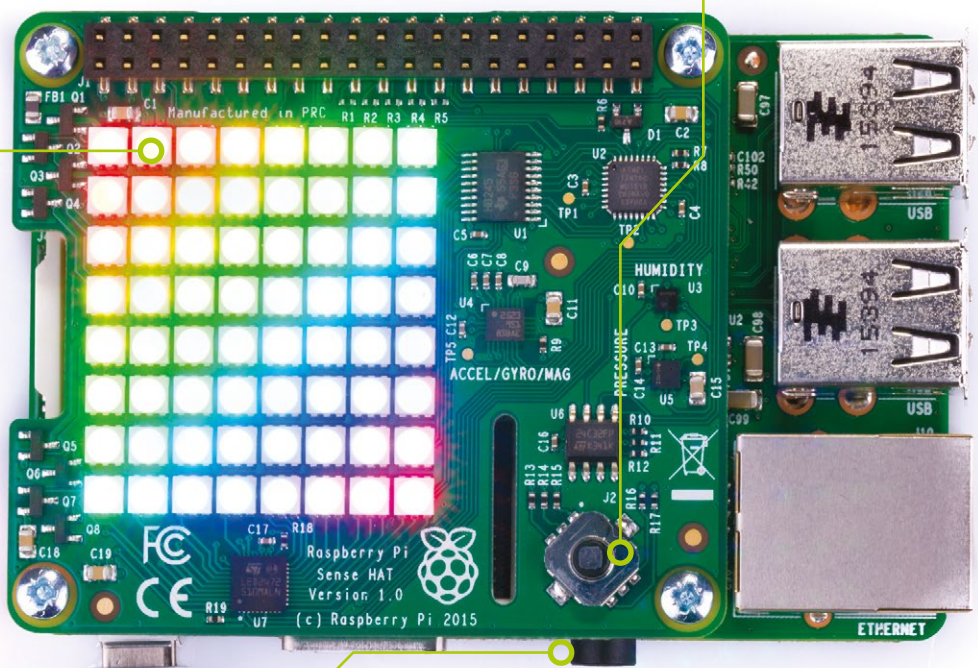
DOWNLOAD THE FULL CODE:
magpi.cc/Nqwqmz



The Sense HAT has 64 LED lights (in an 8×8 matrix) that can be set to any RGB colour. We're using them to scroll text and create a disco effect

The LED matrix is used to scroll track information and produce a disco light effect

Use the joystick to switch tracks and adjust the volume

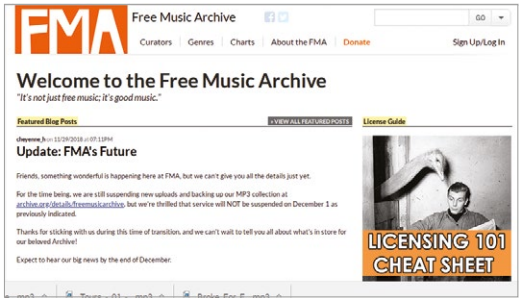


Plug headphones or speakers into the audio socket to hear the music play

Top Tip

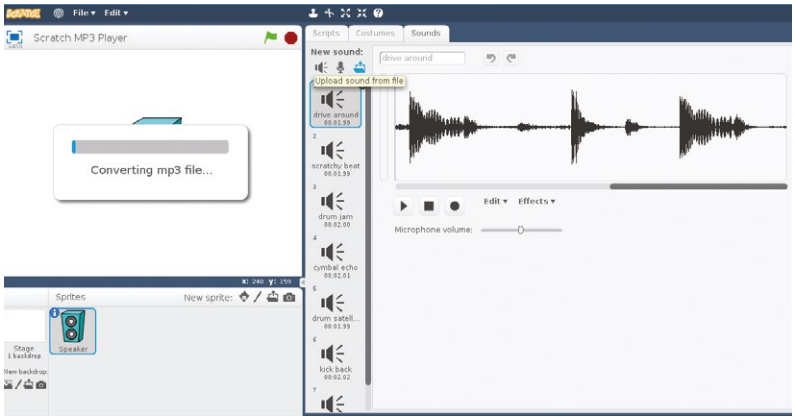
Analogue audio

If you're using headphones or speakers, don't forget to change the audio out on your Raspberry Pi from HDMI to Analog. Right-click on the volume icon on the desktop.

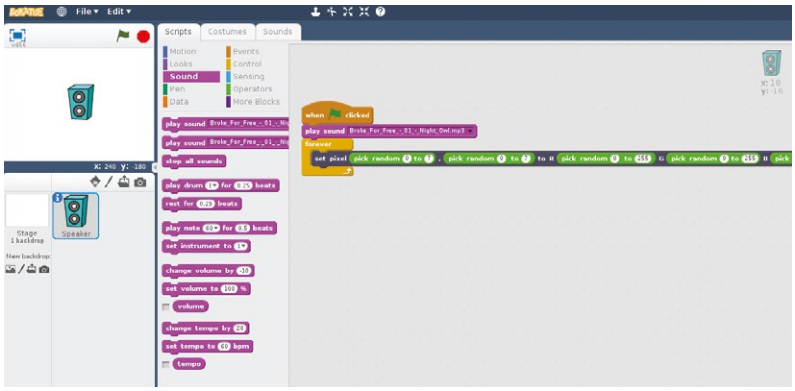


▶ **Step 4** Find some music

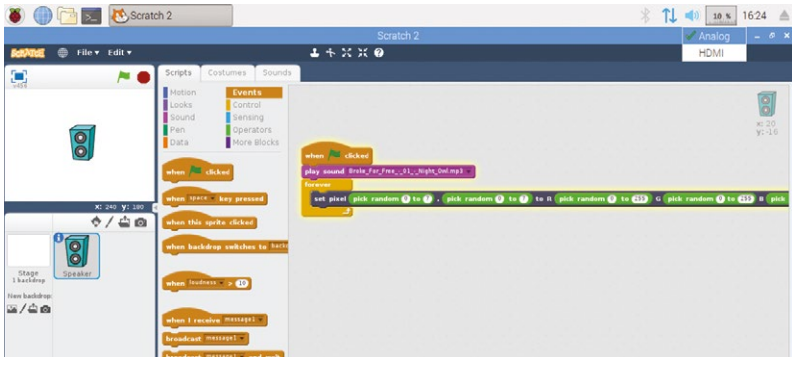
▼ **Step 5** Import the MP3 files



▼ **Step 6** Add a play sound block



▼ **Step 7** Change the Raspberry Pi's audio output to Analog



With the `pick random` block (under Operators), you can set random pixel positions and random colours within the loop to make your Sense HAT sparkle, by choosing LEDs on the Sense HAT to light up (numbers between 0 and 7 on the 8×8 grid), and the amount of red, green, and blue to colour the pixels (numbers between 0 and 255).

Have a go at running your program by clicking on the green flag. You should see the LEDs on the Sense HAT begin to randomly sparkle.

04 Get some tunes

Now you have a disco effect, it's time to make your MP3 player play some tunes! First, save the project. Back in Scratch, click File > Save Project > Create Folder. Name the folder **Scratch MP3** and double-click it to navigate inside. Name the file **Scratch MP3 Player.sb2** and click Save.

> "It's time to make your MP3 player play some tunes"

Gather a few MP3 files together to begin with – you'll want at least three. (We've downloaded three tracks from **freemusicarchive.org**). Save them in the same **Scratch MP3** directory as your Scratch code.

05 Add the sounds

Click on the Sounds tab for your sprite. Get rid of all the stock sounds, such as 'scratchy beat' and 'drum jam' (right-click and select Delete).

Now use the 'Upload sound from file' button to upload all your MP3 files. Choose the MP3 files in your Scratch MP3 folder and click Open.

06 Face the music

Now it's time to play your tracks. Your code should currently look like that in Step 03. You can play a sound using the `play sound` block. Add this block above the `forever` loop.

Click on the green flag to play your track as the disco lights start. Make sure you have speakers or headphones plugged into your Raspberry Pi.

07 Analogue out

At this point you will need to change the sound output to analogue by right-clicking on the speaker icon in the header bar of the desktop and then clicking the Analog option.

Next, you're going to display the number of the track that's currently playing. Click on Data and Make a Variable; call it 'track' (leave it set for all sprites and click OK).

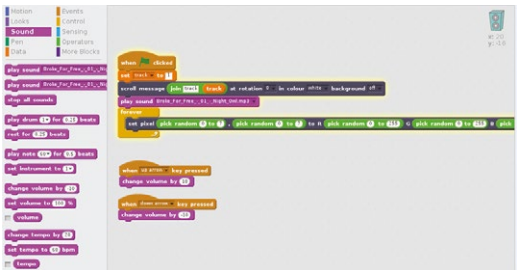
08 Set track

Drag a `set` block above your `play sound` block and choose `track` and 1. Click More Blocks and drag a `scroll message` block under `set track`.

The `scroll message` block can use the `track` variable with a `join` block. Drag a `join` block and enter 'track' in the left field, and drag a `track` variable into the right field (as shown in the image). You can choose whichever colours you like.

Step 8 Scroll the track number on the LED matrix

Step 9 Controlling volume with the joystick



09 Volume control

You can now add some extra controls to your MP3 player. First of all, set up your script so that the Sense HAT joystick can control the volume.

The Raspberry Pi treats the Sense HAT joystick in the same way as the arrow keys on your keyboard. This means that, for example, pushing the joystick up is the same as pushing the up arrow on your keyboard.

Test your scripts again: click the green flag, and then use the Sense HAT joystick to change the volume of your music.

10 Change tracks

Now that you know how to use the joystick, it's time to try to use it to change which track is playing. When the program starts, you will need to stop all sounds.

Use the `when left arrow is pressed` and the `when right arrow is pressed` blocks to `stop all sounds` and to increase or decrease the value of your `track` variable.

You can also use a `scroll message` block to show again which track is being played. You can use the `when arrow is pressed` blocks to change the value stored in your `track` variable.

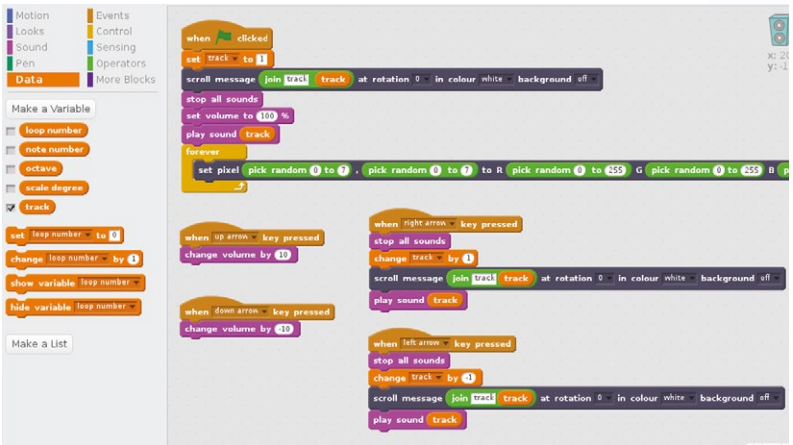
That's it – your MP3 player is complete! Click on the green flag and then have a go at changing your tracks using the Sense HAT joystick.

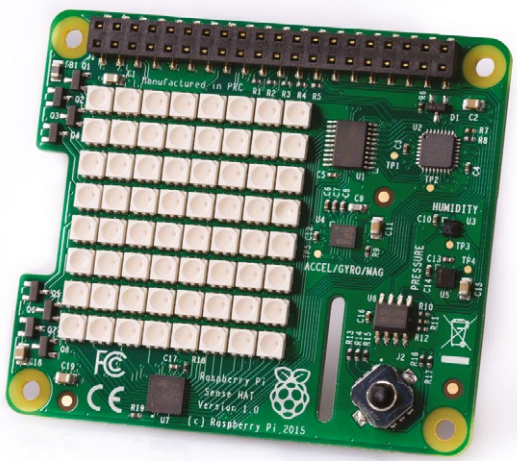
Top Tip

Shuffle

A nice feature to have on an MP3 player is the option to shuffle the tracks so that they play in a random order. Pushing the Sense HAT joystick inwards means the same thing to Scratch as pressing **ENTER** on the keyboard. Update the MP3 player code so that when the joystick is pushed, the code chooses and plays a random track.

Step 10 The complete code


Motion
Looks
Sound
Pen
Data
Events
Control
Sensing
Operators
More Blocks
Make a Variable
loop number
note number
octave
scale degree
track
Make a List
when clicked
set track to 1
scroll message join track track at rotation 0 in colour white background off
stop all sounds
set volume to 100 %
play sound track
forever
set pixel pick random 0 to 7 , pick random 0 to 7 to R pick random 0 to 255 G pick random 0 to 255 B
when up arrow key pressed
change volume by 10
when down arrow key pressed
change volume by -10
when right arrow key pressed
stop all sounds
change track by 1
scroll message join track track at rotation 0 in colour white background off
play sound track
when left arrow key pressed
stop all sounds
change track by -1
scroll message join track track at rotation 0 in colour white background off
play sound track



The Sense HAT is a versatile piece of hardware made by Raspberry Pi. It's called a Sense HAT due to all the sensors on board (hardware and motion), but we're using its other features here

CoderDojo

Can I start a CoderDojo club in my local area?

CoderDojo is a global network of free, volunteer-led, project-based programming clubs for children aged 7–17. Dojos are championed by individuals all around the world who are passionate about giving young people the opportunity to learn to code.

Starting a Dojo is a fun and incredibly rewarding experience

You don't need to possess technical skills to start a Dojo. The most important attribute is that you can bring people together for a shared goal.

We're ready to support you by providing:

- **Learning resources and guides**
- **A free event management system**
- **Certificate templates, digital badges, and giveaways**

"I started a Dojo to give my kids a place to meet other children also interested in programming and making games. I get to see them making new friends, learning from one other, and they loved it. Realising how I had created such a wonderful place for children has ignited a spark in me."

- Maroes, CoderDojo NL

Start your own club. Join us at CoderDojo.com

The CoderDojo Foundation is Irish Registered Charity (524255) CHY20812
and is part of the Raspberry Pi Foundation, UK registered charity 1129409

Part of Raspberry Pi

Social APIs

How do computers communicate over the internet? **PJ Evans** introduces the world of APIs

When we humans want to communicate using the internet, a popular way of doing so is to create a website for others to visit. Being humans, we like shiny things, pretty colours, and anything that makes us go 'ooo' and 'ahh'.

Unsurprisingly, computing devices care nothing for fancy fonts, creative page layouts, or your latest flat-design masterpiece. They want data in an easy-to-read form. So, as we are all clicking to accept cookie policies and dismiss pop-up ads, computers are also chatting away to each other in a much simpler fashion – but instead of 'websites', they are talking using APIs.

An API (application programming interface) is a set of code designed to enable users access a service. Many popular web services have APIs that allow users to plug into their service.

The API provides access to data and actions provided by the host system (the 'server') in the form of functions and returned data. The most common type of API you will encounter will be based on REST or 'Representational State Transfer'. This is a much simpler method of sending instructions and requests to the server. Typically, your response will be in JSON (JavaScript Object Notation).

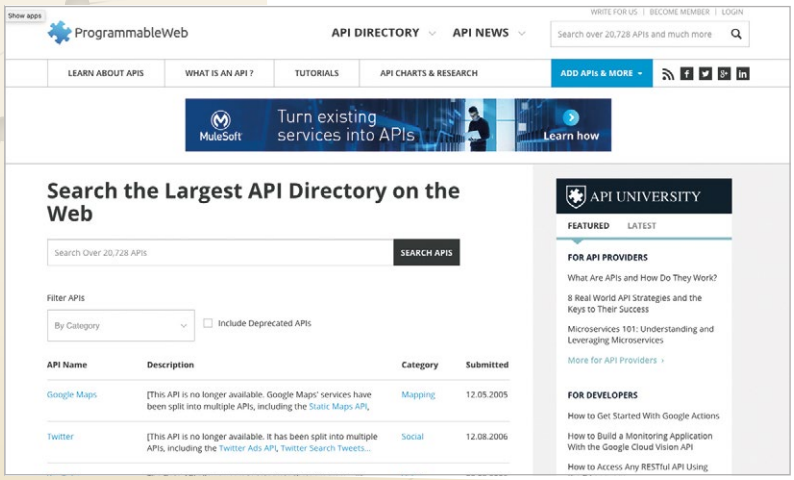
Why API?

So, what are APIs used for? Name any big 'service' on the internet and chances are it will have an API available. When you fire up your Twitter client on your phone, that client-side software is communicating with Twitter using APIs, rather than fetching webpages. This makes things faster and saves a huge amount of bandwidth. In fact, most phone apps that access data from a remote source will be using an API to do it, including your bank, Amazon, Facebook, and so on. Luckily, many of these APIs are opened up for us to use, so fun can be had interfacing an API with, well, anything you like.

We're going to take a look at the Fortnite API, and how to use it to post your wins to social media using a Raspberry Pi.

▲ APIs use the same protocols as the web (HTTP) and return data in plain-text form for the client to parse

▼ There are thousands of APIs for you to explore; **programmableweb.com** provides a directory of the most popular ones



Code a FORTNITE Sense HAT Ticker

Love Fortnite? Don't miss a thing, with the latest scores, news, and events displayed live on a Sense HAT

PJ Evans

PJ is a writer, software engineer, and Milton Keynes Jammer. He is currently hiding from a person who has a tomato for a head.

mrpjevans.com

@mrpjevans

You'll Need

- Sense HAT
- Epic Games user account (**epicgames.com**)
- Internet connection
- Plush Loot Llama (optional!)

Fortnite! Unless you're been on Mars, or living your life in a productive manner, you'll have no doubt have come across this first-person shooter phenomenon. Pitted against 99 other players in a Battle Royale, all you have to do is be the last one standing. As a constantly evolving multiplayer game, there is plenty of data available on player stats, challenges, and items for sale.

Luckily, a simple API provides all this information for us. We're going to use a Sense HAT and Node-RED to create a news ticker for the most dedicated Fortnite players.

01 Prepare your Raspberry Pi

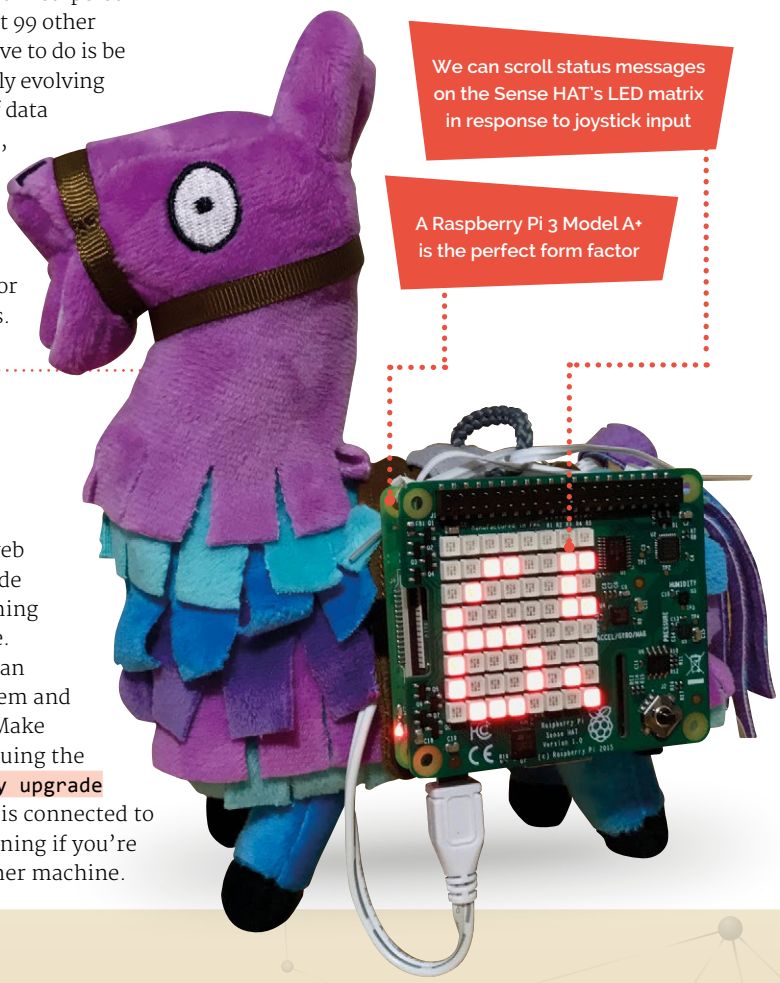
This project will work with any Raspberry Pi model that supports the Sense HAT. We need to use a web browser to create our code, so decide whether you're going to do everything on the Pi itself or another machine. If the latter, you can install Raspbian Stretch Lite as your operating system and skip the graphical user interface. Make sure everything is up to date by issuing the `sudo apt update && sudo apt -y upgrade` Terminal command. Ensure the Pi is connected to the internet and SSH is up and running if you're going to be configuring from another machine.

02 Introducing Node-RED

Our Fortnite device is going to use Node-RED to generate and run our code. This amazing development environment is like Scratch for Internet-of-Things devices and is deceptively powerful. If you've installed 'full' Raspbian with all the recommended software, Node-RED will already be installed and can be found under Programming in the desktop applications menu. Otherwise, follow the command-line instructions at **magpi.cc/eHkkjz** before continuing. Once installed, set Node-RED to always run on boot:

```
sudo systemctl enable nodered.service
```

To test the installation, reboot the Raspberry Pi, then open a web browser and point it to **http://[IP-Address-Of-Your-Pi]:1880** (or **http://localhost:1880** on Raspbian desktop).


We can scroll status messages on the Sense HAT's LED matrix in response to joystick input
A Raspberry Pi 3 Model A+ is the perfect form factor

03 Install Sense HAT node

Programs in Node-RED are known as flows and are made up of 'nodes' that perform specific actions then pass the results of that action (the 'message') to the next node. Node-RED comes with a library of nodes to handle input, output, and steps required in between. To interact with the Sense HAT, we need to install an additional node and some other libraries. Enter the following line-by-line at the Terminal:

```
cd ~/.node-red
sudo apt install -y python-pip sense-hat
sudo pip install pillow
npm install node-red-node-pi-sense-hat
node-red-restart
```

04 Your first flow

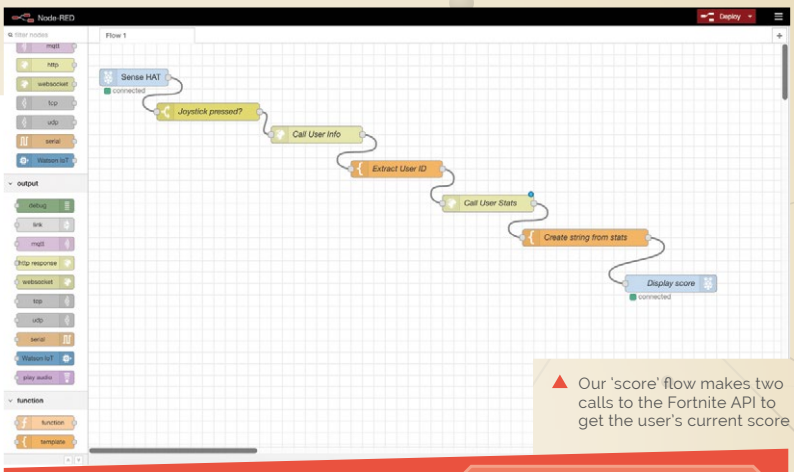
Flows need to start with a trigger: an event that starts the flow. From the left-hand list of nodes, drag 'inject' across to the centre grid (it will change to 'timestamp') then scroll down to the 'Raspberry Pi' section and there should be our Sense HAT. Drag over the 'output' node (small square on the left) and drag a link between the two. Click 'Deploy' followed by the button on the left of the 'timestamp' node. If all is well, you'll see a long number scrolling across the Sense HAT display. Well done, you've written your first Node-RED flow.

05 Connect to the Fortnite API

Although Epic Games does not have an 'official' API, others have provided access for us. One of the most popular is **fortniteapi.com**, which is also one of the easiest to implement. We'll make two HTTP calls to the API to get a user's details, then look up their current score and output it to the Sense HAT. Open up the Node-RED editor as before in your web browser, remove any nodes remaining from the test exercise, then go through the steps in the 'Code Flow' box to create your new flow. Alternatively, download the flow from **magpi.cc/PGvDBa** and use Menu > Import > Clipboard to load the flow in.

06 Add more functions

Now you can view a user's score just by pressing the joystick button. However, Node-RED flows don't have to be linear, so you can process several actions to produce different reactions and branch code based on the incoming data. Download the 'advanced' flow from **magpi.cc/MjeNBe** and import using Menu > Import > Clipboard. Have a look at how each step is configured and see how you could change things to add new features or enhance existing displays. Don't forget, you can use Node-RED to talk to thousands of different APIs!



▲ Our 'score' flow makes two calls to the Fortnite API to get the user's current score

score.json

DOWNLOAD THE FULL CODE:
magpi.cc/MjeNBe

› Language: **Node-RED**

To create your Fortnite flow, go through each step below, drag the required node onto the grid, link it to the previous node moving from left to right. Set the properties by double-clicking on the node and completing the form fields as given. Leave all other fields at their default settings.

Sense HAT Input
Output: Joystick events only

Switch
Name: Joystick pressed?
Property drop-down: Expression
Property field:
`payload[key='ENTER'] and payload[state=0]`

HTTP Request
URL: https://fortnite-public-api.theapinetwork.com/prod09/users/id?username=USERNAME
(N.B. Replace USERNAME with your Epic Games account name)
Return: a parsed JSON object
Name: Call User Info

Template
Name: Extract User ID
Set property: url (leave drop-down as 'msg')
Template:
`https://fortnite-public-api.theapinetwork.com/prod09/users/public/br_stats?user_id={{payload.uid}}&platform=pc`

HTTP Request
Return: a parsed JSON object
Name: Call User Stats

Template
Name: Create string from stats
Template:
`Solo Score: {{payload.stats.score_solo}}`

Sense HAT Output
Name: Display score

Public API projects to check out

Publicly available APIs provide access to a bewildering array of information. Here are some of our favourite Raspberry Pi-based projects

FETCHING THE WEATHER

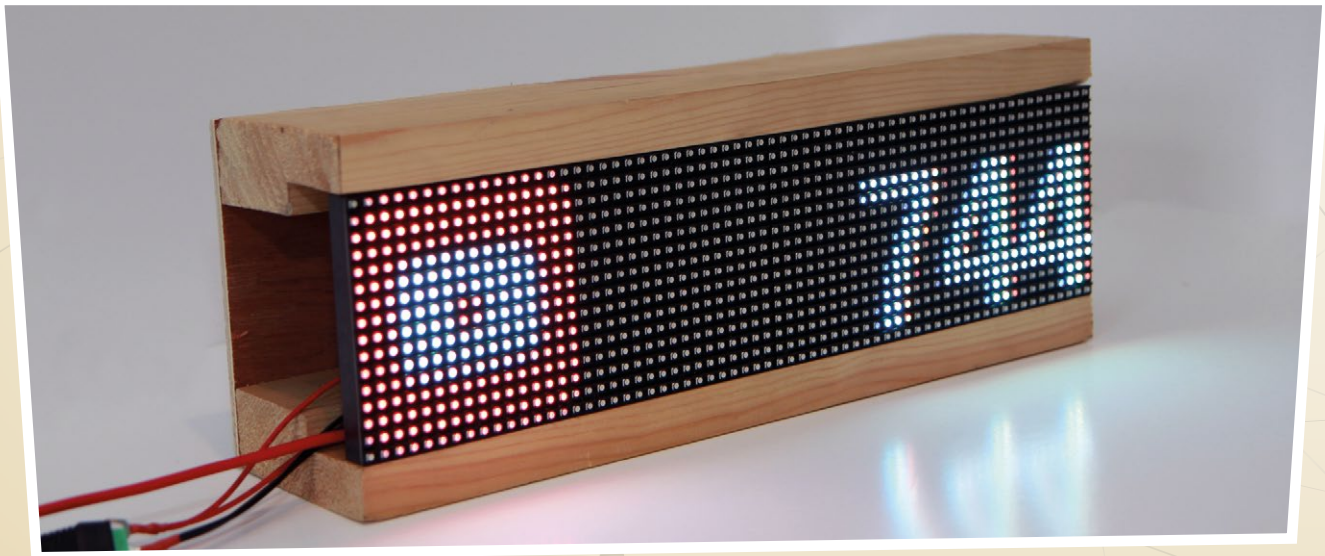
This tutorial is great way to get comfortable with programming APIs. Using an API that provides access to school weather stations, you can fetch live weather data in just a few lines of Python. In what fancy way could you display the results?

> magpi.cc/FvdhbK

SOCIAL MEDIA FOLLOWER COUNTER

Here's something a bit more advanced, but it does show that connecting APIs can be beautiful as well as practical. This project involves building a large LED matrix using two displays, to show a rotating social media follower count using multiple APIs. Ideal for an office or just bragging rights at home.

> magpi.cc/WhVLNA



AIY VOICE KIT

Another great benefit of APIs is the ability to offload difficult work to a server 'somewhere in the cloud'. Google's AIY Voice Kit uploads audio from the microphone to Google's servers via an API, to undertake all the difficult parsing that would overwhelm a Raspberry Pi.

> aiyprojects.withgoogle.com

Image Credit: Lucas Berbesson

NOTIFICATION BOX

This pretty box lights up when a notification occurs on YouTube, Facebook, or Instagram. The light colour changes based on which service sends the notification, and as a bonus you get a round of applause! Again, a great use of amalgamating multiple APIs into a single device.

> magpi.cc/uOnUvG

Glossary

Like most technology, the world of APIs is filled with Jargon. PJ Evans is here to help

REST

'Representational State Transfer'. A de facto standard for how APIs work. Commands are issued over HTTP using the verbs GET, PUT, POST, PATCH, and DELETE to determine the type of action.

SOAP

'Simple Object Access Protocol'. An older system for issuing commands, championed by Microsoft.

JSON

'JavaScript Object Notation'. The most common way data is formatted in API transactions. Think of it as a more advanced version of CSV files that can handle nested data. See **json.org** for more details.

YAML

'Yet Another Markup Language' or 'YAML Ain't a Markup Language'. YAML is a competitor to JSON that is even easier for humans to compose and read. Like Python, it relies on indentation of code.

XML

'eXtensible Markup Language'. A method of representing data that is platform independent. Although JSON and YAML are the young pretenders, XML is still very popular and is a sibling of HTML.

ENDPOINT

When referring to an API, the 'endpoint' is the base URL used to access it. Different API resources are additional paths from the endpoint.

API/USER KEY

In order to provide security, an API will often require that both an API key and a user key are supplied along with the request. The user identifies your account and the API key represents the application itself.

OAUTH

A far superior form of security is OAuth 2. This requires that the server gets explicit permission from the user before allowing access to the API client. Upon approval, typically a manual process, the client is given a 'token' which is used for subsequent access.

OAUTH TOKEN

If an API implements OAuth 2 security, the API client must use its unique and secret key (issues at time of approval) to get a 'token': a string which acts as a password and is unique to that session.

TRANSIT SIGN

Bus shelter and train platform signage uses APIs to receive messages to display. Often those APIs are available to the public. Paul Shved decided that rather than a magic mirror or pop-up display, he would really go for it, so he now has his own full-size signage at home.

> magpi.cc/ovSjxk

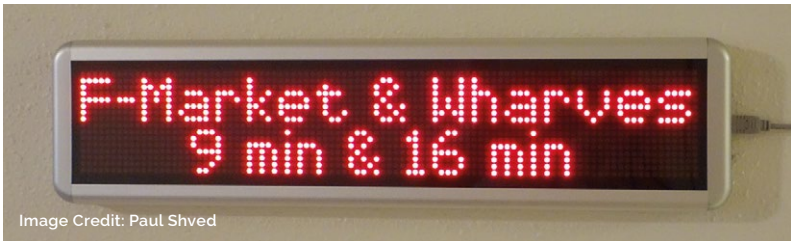


Image Credit: Paul Shved

TinyPi Pro

▶ Pi0cket ▶ **pi0cket.com** ▶ £69 / $87

The smallest retro game console you ever saw, and you can make it yourself. **Rob Zwetsloot** provides a Pi Zero for one

SPECS

DISPLAY:
240×240 pixels, 1.3-inch

BATTERY:
Rechargeable 400 mAh

DIMENSIONS:
69 × 34 × 20 mm

PORTS:
HDMI out, USB in

STORAGE:
Swappable microSD card

We first looked at the TinyPi – originally called the Piocket – way back in *The MagPi* #59 (**magpi.cc/59**) when it was just an interesting little project. Now the maker, Pete Barker, has put together a little kit so you can make your own extremely small game system.

And we do mean small – it's designed to be the same size as a Pi Zero, albeit just shy of 20 mm deep. It's a pretty remarkable kit in that sense: fitting a D-pad, six face buttons, and two shoulder buttons, while also including a screen and speaker in the chassis is both impressive and a tight squeeze.

As mentioned above, it does come as a kit, and you need to supply your own Pi Zero for it (GPIO pins not required), as well as a microSD card to install an operating system to.

For a project kit this small, you'd usually be required to do a little soldering yourself to ensure everything fits as intended. Not so with the Tiny Pi – in fact, the only fastening you need to do is with eight screws connecting to four spacers, and an Allen key to tighten them is supplied in the kit.

Tiny build

With something this small, you also might imagine it to be quite fiddly – and the tweezers included with the kit will hardly allay those fears. However, the only really fiddly part was attaching the (quite small) battery. The tweezers worked perfectly for that, while everything else just dropped, slotted, or clipped into place.

The packaged instructions make the build look simple, and it mostly is: unfortunately, a couple

▲ The kit is nicely laid out, with everything in plain view

of bits are neglected in the explanation, although we're assured that there will be very fleshed out online resources by the time its released, as well as some tweaks to the included instructions.

Tiny power

Thanks to a combination of RetroPie and the power of a Pi Zero, the TinyPi Pro has the means (and oomph) to play a wide range of emulated and homebrew games. The RetroPie setup is easy, but you need to know how to skip the controller configuration once all the available buttons are used (tip: hold down any button).

While there are more than enough buttons for the majority of games you'll be able to play on the TinyPi Pro, actually using them all with clumsy adult mitts for some games is quite difficult. Fingers and thumbs get in the way of each other, and you'll soon get a cramped hand from holding it in a way needed to reach the shoulder buttons. We found games that didn't need that many buttons worked a lot better, but kids with smaller hands might be fine with the full set.

The small screen is fine, though, and you'll be hard-pressed to find a game where you'll struggle to make out everything that actually runs on the Pi Zero.

It's a lovely kit with a fun, quick build. The final result is impressive, but it's not the very best thing to play games on. Still, it fits neatly in a pocket if you really need a retro hit out and about.

▲ Compared to its contemporaries, it's an incredibly small device

◀ While the size is very convenient, it does mean it's a little awkward to play

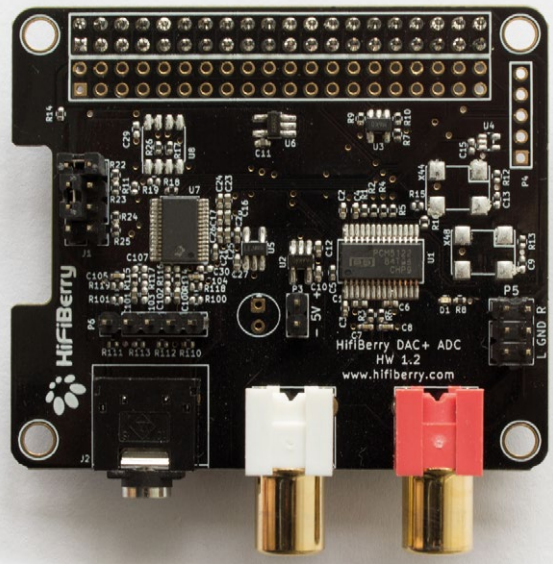
> We found games that didn't need that many buttons worked a lot better

▲ Much smaller than a banana, it's not quite so easy to hold

Verdict

A really neat kit that has a fun build and an amazing final product. It's a little too small for giant adult fingers, but may be better suited to younger players.

8/10



The DAC+ ADC is a HAT that simply slides onto the Pi's GPIO header, with no further connections required

HiFiBerry DAC+ ADC

HiFiBerry | magpi.cc/NshChD | £45 / $57

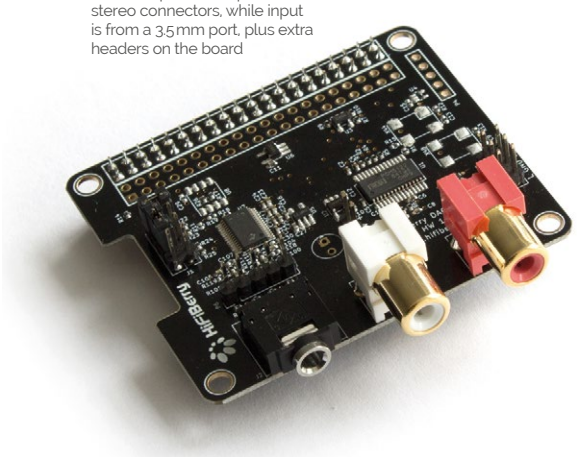
There are lots of audio output HATs for the Pi, explains **K.G. Orphanides**, but this one also records, thanks to an integrated analogue-to-digital converter

Unlike most Raspberry Pi audio HATs, the HiFiBerry DAC+ ADC features an analogue audio input, so you can record, as well as play, sound. That's perfect for compact audio production projects.

Physical setup is easy, but you'll need a Linux 4.18.12 kernel to use the analogue-to-digital converter (ADC) hardware. For full kernel upgrade and boot configuration instructions, see the DAC+ ADC data sheet on HiFiBerry's site.

The Burr-Brown PCM5122 DAC chip is a popular choice for reasonably priced computer audio hardware, and with good reason. It's a pleasure to listen to and, via a stereo RCA output, really sings through high-quality speakers. There's no on-board headphone amp, though.

Audio output is via a pair of RCA stereo connectors, while input is from a 3.5mm port, plus extra headers on the board



The ADC – a Burr-Brown PCM1861 – has a 3.5mm stereo input. This is, by default, configured to accept line-level audio, such as you'd get out of your mobile phone or the line-out connectors on most audio gear. You can use it to digitise analogue media such as cassette tapes, turn your Pi into a portable instrument effects box, or record from a mixing desk.

> Use the ADC to digitise cassette tapes, turn your Pi into a portable instrument effects box, or record from a mixing desk

Singing its praises

You can also connect dynamic microphones – we tested this with a Shure SM58 – if you adjust a jumper switch to enable 32dB gain. It works perfectly for vocal recording, karaoke parties, or even enabling a software audio passthrough to make an improvised public address system.

Note that the DAC+ ADC can't provide the phantom power required by either studio-grade 'true' condenser mics or compact electret mics. Additional headers on the board allow you to hook up external amplifiers and balanced inputs.

SPECS

HARDWARE:
DAC: Burr-Brown PCM5122 (192kHz/24-bit), ADC: Burr-Brown PCM1861 (192kHz/24-bit)

SNR:
DAC: 112dB typical, ADC: 110dB typical

PORTS:
3.5mm audio in, stereo RCA out, balanced input header, output header

Verdict

The DAC+ ADC handles lossless music, games, MIDI soft synths, and sound production brilliantly. If you want a compact board to add full audio functionality to your Pi, this is it.

9/10

THE OFFICIAL Raspberry Pi Beginner's Guide

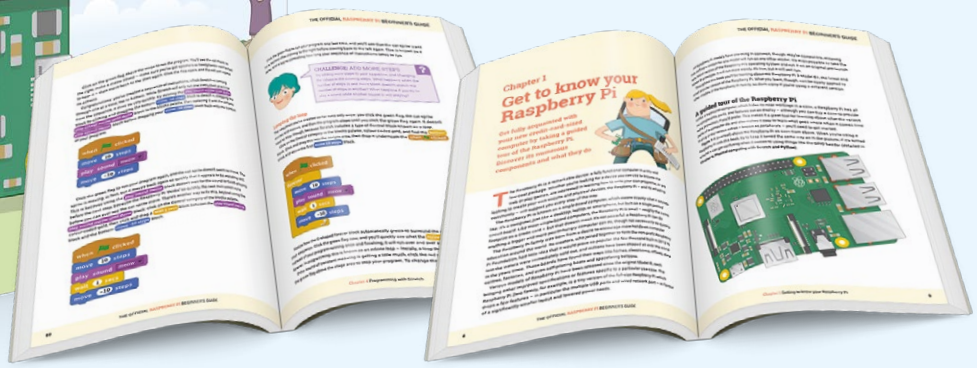
The only guide you need to get started with Raspberry Pi

Inside:

- Learn how to set up the Raspberry Pi, install an operating system, and start using it
- Follow step-by-step guides to code your own animations and games, using both the Scratch and Python languages
- Create amazing projects by connecting electronic components to the Pi's GPIO pins

Plus much, much more!

£10 with FREE worldwide delivery



Buy online: magpi.cc/BGbook

NEW

Wireframe

Join us as we lift the lid on video games

Visit **wfmag.cc** to learn more

Inky wHAT

▶ Pimoroni ▶ **magpi.cc/gLoZJW** ▶ £48 / $51

This larger e-ink display offers more room for images and text. By **Phil King**

SPECS

DISPLAY:
4.2-inch EPD, 400×300 pixels

USABLE DISPLAY AREA:
84.8×63.6 mm

WORKS ON:
Any 40-pin Raspberry Pi

What's a wHAT? A wide HAT, that's what. The Inky wHAT's 4.2-inch screen has 4.7 times the display area of the Inky pHAT, and 5.4 times the number of pixels. So it's far better for displaying detailed images and longer text. A small female header on the rear breaks out ten GPIO pins, including I²C and SPI – handy for attaching sensors and other components.

Mounted display

We tested the black/red/white version (it's also available in black/white). As the wHAT comes fully assembled, you just need to mount it on your Pi using the supplied metal stand-offs, and a GPIO header extender to boost the height for full-size Pi models. Care must be taken not to put too much force on the fragile glass screen, however.

A single command installs the Inky Python library and code examples, including a couple specifically for the larger screen. A famous quotes program shows how to reflow text to fit the display, while an image dithering script converts any picture by resizing and changing the colour palette; the latter works best with simple graphics – photos are still recognisable but a bit grainy. The clearest results come from creating your own images in GIMP, using an indexed colour palette.

> It takes around 25 seconds to do a full screen refresh, so it's only suitable for certain projects

It takes around 25 seconds to do a full screen refresh: there's a fair bit of flickering, then the images gradually appear. So it's only really suitable for projects needing an occasional screen update, but – as an e-ink display – only uses any power while refreshing.

Verdict

Due to the lengthy screen refresh time, it's not usable for every project. However, its larger display is ideal for a weather report, calendar, news headlines, or to-do list.

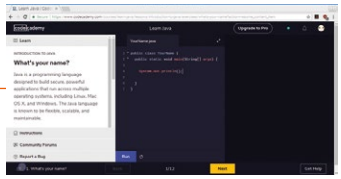
8/10

▼ Dwarfing the pHAT, the Inky wHAT offers a lot more screen estate

▲ The extra header on the rear breaks out ten of the Pi's GPIO pins

Everything you need to learn Java with Raspberry Pi

It's clunky, powerful, and divides opinion faster than Gordon Ramsay chopping an onion. It's time to learn Java. By **Lucy Hattersley**



Codecademy: Learn Java

AUTHOR

Codecademy

Price:
$20 (per month)

magpi.cc/StpvZf

Java isn't like other languages, and its complex and wordy syntax can be baffling for beginners. There's no denying its power and popularity, though.

Claiming over 25 million users, the Codecademy website has courses (many free) on a plethora of different programming languages and tools.

Learn Java is its flagship course, but sadly not one of the free ones. You can follow some introductory lessons, but you'll need to get the $20 a month Codecademy Pro plan to finish it.

It's worth it, though. On Codecademy you'll find a comprehensive stack of tutorials that cover all the basics from the Java syntax, through to creating objects and data structures.

You learn via interactive lessons that run inside the Chromium browser. The window is split into three columns: on the left is the lesson text; in the middle is a code editor; on the right is an emulated terminal. Typically you'll find boilerplate code in the editor and have to add elements by following the instructions. Click Run and check the output in the terminal.

Codecademy is superb at hand-holding, and the web-based setup is ideal for beginners. This is especially handy given Java's obtuse nature.

The best books

Read up on Java

JAVA, A BEGINNER'S GUIDE

Written by Java guru Herbert Schildt and published by Oracle Press, this feels to us like the closest thing to an official guide.

▶ magpi.cc/jfZUCy

THINKING IN JAVA

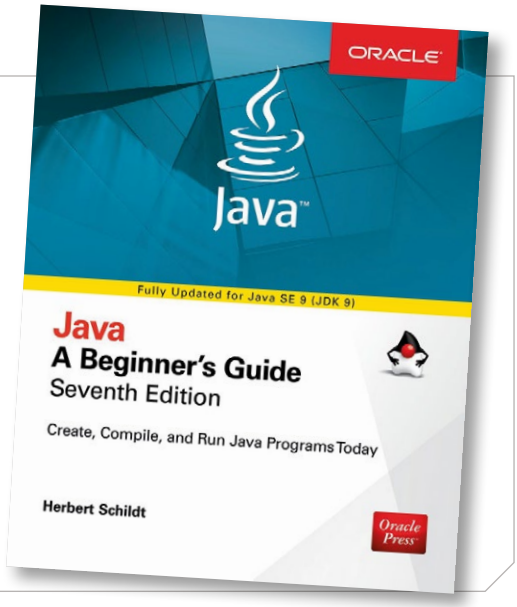
Once you get past the nuts and bolts, it's time to start thinking about how to improve as a programmer. Java is a useful language for thinking about coding concepts, and this book is the ideal partner for the curious coder.

▶ magpi.cc/cJegrn

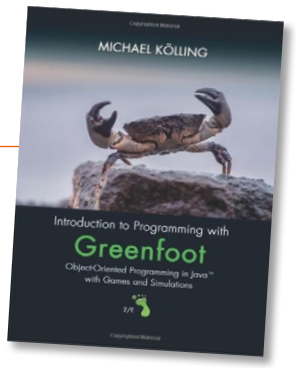
HEAD FIRST JAVA

If you only buy one book on Java, then make sure it's this one. As well as covering the basics, it dives deep into esoteric concepts such as compiling and the JVM (Java Virtual Machine). All aspects of OOP are covered.

▶ magpi.cc/BLJrOf



Introduction to Programming with Greenfoot



AUTHOR

Michael Kölling

Price:
£86 / $102

greenfoot.org/book

Greenfoot is a Java IDE developed by Michael Kölling from King's College London. On the Pi you'll find it under Programming in the Raspbian menu.

The Greenfoot IDE creates a visual world for your Java classes to live in, much like Scratch or Blockly. Students create 'actors' which live in this world, and use it to build games, simulations, and other graphical programs.

This visual nature enables Greenfoot to leapfrog the abstract nature of programming, with students able to see their objects interacting in a visual manner. Its actors and worlds are coded in standard Java. So you get a good combination of standard programming and visual execution.

This book has fun projects to follow, including building a piano, simulating bubbles and leaves floating in the air, foxes and rabbits, and a simulation of the motion of stars and planets.

Make sure you download the correct project files for the Pi and its Greenfoot IDE 2.4 – they're marked 'Old version of book scenarios for Greenfoot 2.x' on the book's webpage.

edX

AUTHOR

edX

Price:
Free

edx.org/learn/java

If you want to take your Java learning to a professional level, check out the selection of MOOCs (massive open online courses) taking place on edX.

Intended as university-grade courses, MOOCs are challenging but rewarding and you get to share the experience with other learners online.

The Java courses are provided by a range of different institutions, including Microsoft, Red Hat, and UBC. MIT has also offered courses in the past, although none of its Java software development courses are running at the moment.

Many courses are self-paced, but some follow a strict timetable with projects and final exams. You'll find a dedicated bunch of individuals in the forums and keen staff willing to nudge on newcomers.

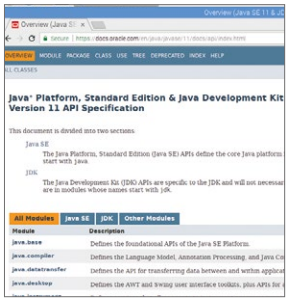
The courses are definitely a challenge, and you can pay to have a verified certificate. However, you can enrol on most courses for free and often the learning is worth it regardless.

We've found the estimated effort to be lowballed. So be prepared to put in more hours than course providers suggest.

While edX might not be the best place to start, it's the place to head after you've got some grounding in programming and want a challenge.

Interactive Websites

Bookmark these sites as soon as you begin your Java journey



JAVA API

The Java API is one of the best maintained set of documents in existence. Every object and class you encounter is detailed in a standardised manner.

▶ magpi.cc/kBTusc

LEETCODE

LeetCode has a web-based Playground IDE that you can use to quickly test out Java code (along with many other languages). But it also has a bunch of articles, interviews, and opinion pieces that are fun to read.

▶ leetcode.com

W3SCHOOLS

This website is better known for its web developer support, but Java is a popular language amongst web developers so it has a comprehensive Java section. There are some neat tutorials and example Java code that is handy to have around.

▶ w3schools.com/java

10 Best: Wearables

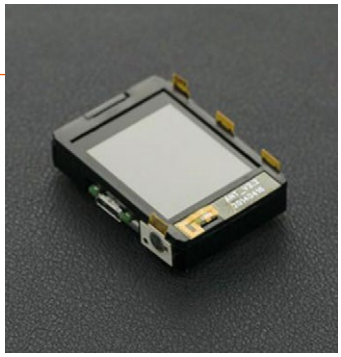
These devices and kits are amazing ways to make your aesthetic electric

Over the years, people have put Raspberry Pi computers in almost everything. That also includes clothing and devices to keep on your person, creating what are called wearables. You can now get kits and products that act as wearables, allowing you to upgrade your fashion sense with a bit of Pi.

Mixtile GENA

If you don't fancy the current selection of smartwatches, or have big ideas for a very small screen, the Mixtile GENA is a great little screen to have a look at. It's also bristling with sensors, and can connect wirelessly via Bluetooth.

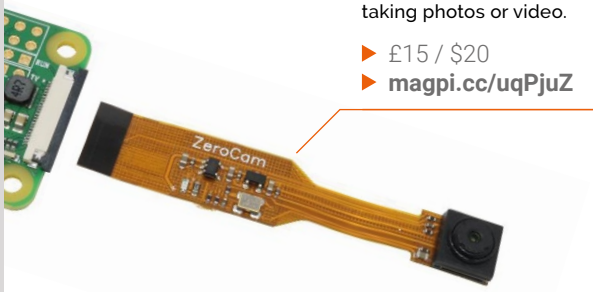
- £37 / $48
- magpi.cc/PipfqR

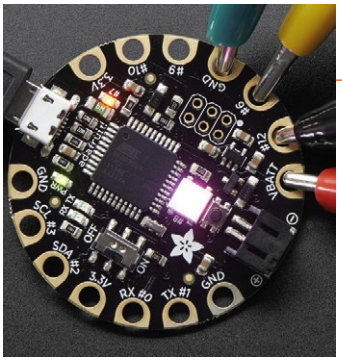


ZeroCam

Need the smallest camera possible to do fun or secret stuff on your person? The ZeroCam lets you easily affix a very small, Pi-compatible camera to your clothes or wearable project for computer vision or just taking photos or video.

- £15 / $20
- magpi.cc/uqPjuZ

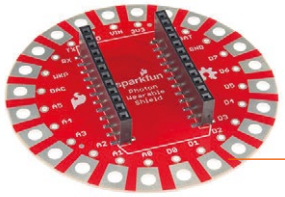


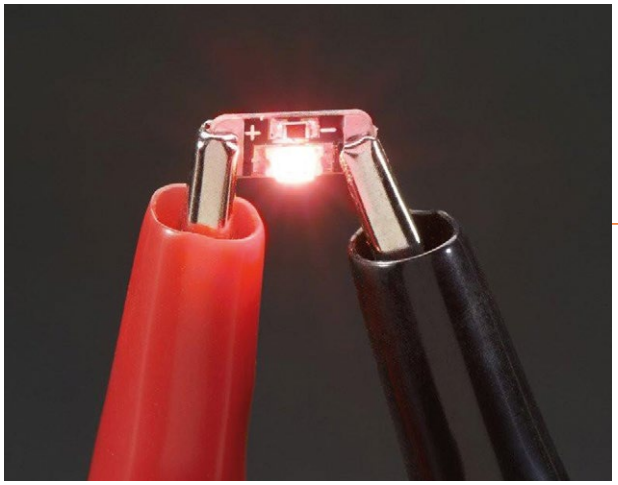


Adafruit FLORA

The FLORA range is the main programmable wearable range from Adafruit, with big holes perfect for conductive thread and programmable as an Arduino device to control other wearable bits, along with its own light and on/off switch.

- £14 / $15
- magpi.cc/GZwBGP





LED Sequins

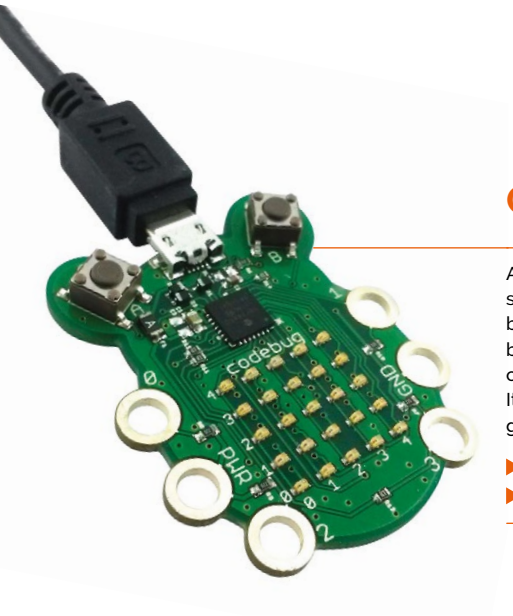
Adafruit does a wide range of different wearable tech, but these LED sequins are the simplest of the bunch. Just give them a bit of power and they'll light up – perfect for fun clothes or awesome cosplay effects.

- £4 / $4
- magpi.cc/cTxdRC

Photon Wearable Shield

This shield lets you use the SparkFun Photon IoT board with your clothes – the breakout holes are perfect for conductive thread – and thus enables your coat to become a walking Internet of Things.

- £6 / $6
- magpi.cc/rGGJri



CodeBug

A nice, fun, simple board that is shaped like, and easily programmed by, a Raspberry Pi. It has two buttons, a matrix of LEDs, and connectors for conductive thread. It's a great starter piece for those getting into wearables.

- £15 / $17
- **codebug.org.uk**

Bearables

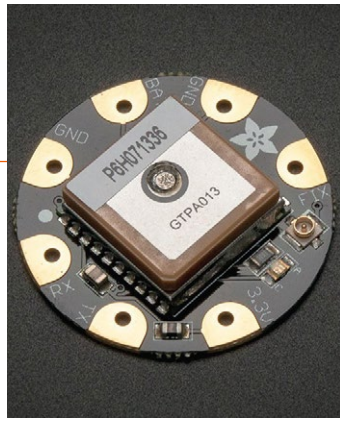
This is a really fun, quickish project that lets you add a cute, light-up animal to your clothes. There's extra stuff you can do with it, like add a light sensor and such, and it comes with some conductive thread. Perfect for younger makers.

- £13 / $13
- **magpi.cc/dqGmfL**

FLORA Ultimate Wearable GPS Module

While there are many great FLORA modules, we just wanted to highlight this tiny GPS one. We have visions of creating a backpack that will light LED direction indicators during your bike ride to work.

- £37 / $40
- **magpi.cc/tMCkTJ**



LilyPad Button

There's an entire LilyPad wearable range with different bits and pieces. It's low profile with no sharp corners, useful for clothing that might move over you. There's also slide switches, LED boards, and more in the range.

- £2 / $2
- **magpi.cc/wnBaTQ**

Electric Paint

If you can't use conductive thread, you can always try some conductive paint, hidden among other colours or out of view. Bare Conductive also has a range of electronics that are small and work well with Pi and this paint.

- £7 / $9
- **magpi.cc/muEGwh**

WEARABLE PROJECTS

Need some inspiration for what to make? Raspberry Pi has a simple getting started guide (**magpi.cc/2B9p66L**) and you can find loads of excellent builds on Instructables, Hackster, and other project sites.

Lisa Rode

Bringing robots into the classroom makes for fun lessons that teach a lot more than computer science

> Category Educator | > Day job Teacher | > Twitter @roderunners

The Raspberry Pi Foundation has done amazing work helping to reform the computing curriculum at UK schools. People in other countries around the world have taken notice, especially those involved in education. One of those educators is sixth grade teacher Lisa Rode from Springfield, Virginia.

"In 2014, I started a Raspberry Pi after-school robotics club using the Dexter Industries GoPiGo robotics kit to teach students programming, engineering, and problem-solving skills," Lisa tells us. "After seeing the power of teaching physical computing and computational thinking skills, I began to integrate more technology and robotics into core content lessons. Over the past four years, I have transformed my classroom into a makerspace where all students are given opportunities to show their understanding in multiple ways. The use of robotics, programming, 3D modelling and printing, and other resources are accessible to all students for use in core content lessons, as well as student-led projects."

Lisa teaches maths, social studies, science, reading, and writing as part of these core content lessons, and has been teaching in a very diverse and inclusive classroom for the

▶ Programming GoPiGo robots using the Bloxter language

▼ Several Raspberry Pi workstations are set up for Lisa to teach on

▲ Teaching American history – in this case, the Westward expansion – with robots

past few years. Inspired by Picademy, she created a course for teachers at her school to learn about programming and physical computing, and also works with Dexter Industries on their curriculum.

What's your involvement in Dexter's curriculum?

I write a curriculum for the Dexter Industries robotics kits. Much of my work is making lessons that are integrated into core content areas such as lessons on plot structure where students turn the GoPiGo robot into a character and program it to act out a story to include all of the main plot elements (exposition, rising action, climax, and falling action). Another set of lessons that integrates core content concepts is the Mars Rover unit where students learn about space exploration, space technology, and planets in our solar system. They then design their own rover using the GoPiGo as a starting point.

> "Much of my work is making lessons integrated into core content areas"

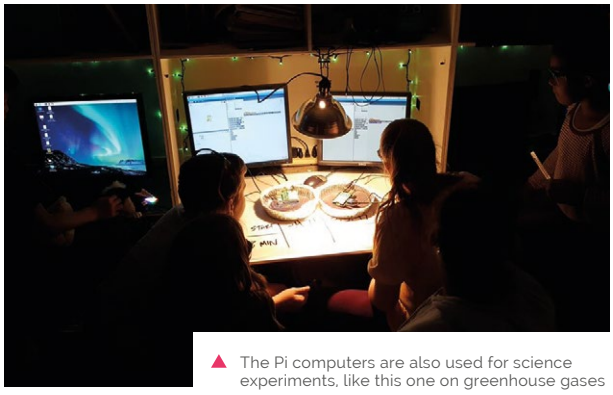
I also work with Nicole Parrot, the CTO at Dexter Industries, to discuss new blocks in Bloxter (one of the programming languages for the GoPiGo robots) to help fulfil curriculum needs. After testing these blocks on my own and in my classroom, they become available for everyone to use. One example is a 'curve' block that was created after my students created models of the movement of Earth around the sun using GoPiGo robots.

What's your background in computing?

I do not have any formal education or background in computing. I came across information on the Raspberry Pi a couple of years ago over the summer and decided to purchase one for myself. After tinkering with it on my own, I thought that it would be great to have a couple set up in my classroom. Over time, that led to even more Raspberry Pi workstations in my room and a set of GoPiGo robotics kits.

Over this past summer I was fortunate to be able to participate in Picademy in Jersey City. I learned so much and brought that information back to my school.

▼ A lot of fun robotics get taught in Lisa's classroom



▲ The Pi computers are also used for science experiments, like this one on greenhouse gases

Project highlights

PLANETARY ROVER

"One of my favourite projects is when the students transform their GoPiGo robots into planetary rovers. This year we added a little twist. Since NASA lost contact with the Mars Opportunity rover in June due to a planet-wide dust storm, part of the mission for each rover team was to help save Opportunity. Student teams worked to build and program their rovers with the mission of finding and rescuing Opportunity."

CARDBOARD ARCADE

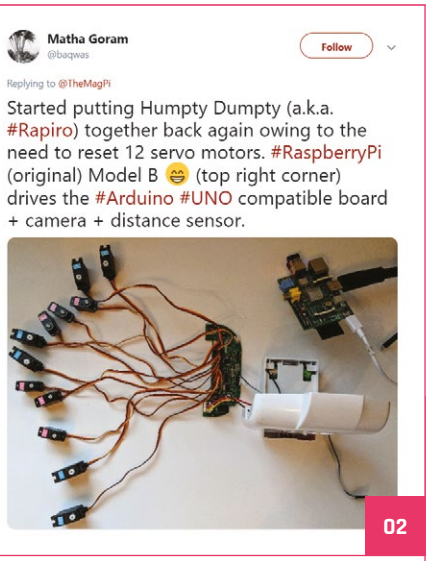
"A few years ago, my students created a cardboard arcade game similar to Pac-Man that was powered by a Raspberry Pi and programmed in Scratch. The controls for the game were created using a Makey Makey."

This Month in **Raspberry Pi**

New Year, **New Projects**

What were you all working on over the break?

The Christmas period usually means time off for a lot of people, and many prezzies under the tree that can work with a Raspberry Pi. With free time and new toys, it's very conducive to digital making.


Matha Goram
@baqwas
Follow
Replying to @TheMagPi
Started putting Humpty Dumpty (a.k.a. #Rapiro) together back again owing to the need to reset 12 servo motors. #RaspberryPi (original) Model B (top right corner) drives the #Arduino #UNO compatible board + camera + distance sensor.

02

Pi 0 in your Pocket
@Pi0CKET
Replying to @TheMagPi
I may or may not have been playing with the keyboard driver for clicker
#Pi0CKET

01

01. A future upgrade for the TinyPi Pro?
02. The Rapiro robot is a favourite of ours, but it can be a bit finicky
03. Rosie is the *Countdown* robot that we featured in *The MagPi* #66
04. Pi Wars is slowly creeping up… are you taking part?
05. We love this music player powered by Pi
06. Definitely take a look at this project!
07. An easier way to track your BTC value?

Rosie the Red Robot
@rosie_red_robot
Following

Replying to @TheMagPi

If #Aliens arrive in 2019 (and they are kind enough to let us know their GPS coordinates) we'll be ready with our tracker thingy.

Rosie the Red Robot @rosie_red_robot
[Video] Who wants to feel like they are being surrounded by #Aliens? 👽 Anyone? #RaspberryPi GPS tracker / compass 🧦 with static landmarks. #Xenomorph optional. #HTML5 #JavaScript #MQTT #SocketIO

Show this thread

03

Laurence Molloy
@MolloyLaurence
Follow

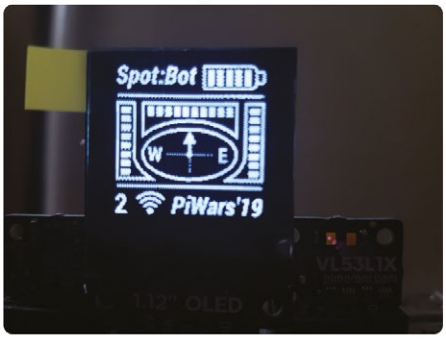
04

Replying to @TheMagPi

Been working on a #PiWars robot display screen, using @Pimoroni breakout garden OLED.

Current design contains readings for 3x ToF sensors, compass bearing, battery level, mode & comms status in 128x128 px.

Will be controlled via a standalone process, updatable via message API.



Chris Patty
@ChrisJPatty
Follow

This is what it looks like inside. There's a raspberry pi that runs the software, and then just a card reader in the lid.

05

Marty
@KutilDomaci
Follow

07

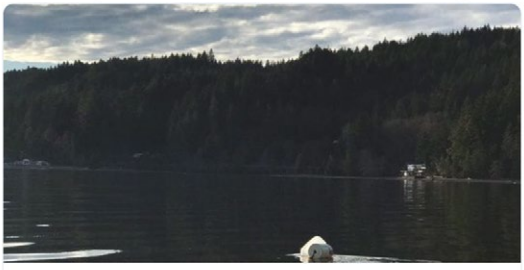
Replying to @TheMagPi

Yep! I picked up this project from my drawer, it´s a Dashboard Pi kind of. Still deciding for what it can be used. Currently the most favorite is weather dashboard.

Steve Haan
@HaanHouse
Follow

06

@TheMagPi I'm a little late but My #PIZero spent Sat. at 111m underwater in Hood Canal (Washington) for 2 1/2hrs. FOLLOW along



Open Explorer | Southern Salish Sea ROV BRUV Exploration
Hood Canal just south of Eagle Creek @ 111m (364'). Prawn, Sole, Arrow worms, and lots of Plankton coming to visit BRUV. Impressive clouds of plankton with many
openexplorer.nationalgeographic.com

Scratch Conference Europe hosted by Raspberry Pi Foundation!

The first-ever European Scratch Conference will be held in Cambridge this summer

The Raspberry Pi Foundation has announced that it will host the first-ever Scratch Conference Europe in the UK this summer: from Friday 23 to Sunday 25 August at Churchill College, Cambridge.

"Scratch Conference is a participatory event that gives hundreds of educators the chance to explore the creative ways in which people are programming and learning with Scratch," Helen Drury, Events and Outreach manager at Raspberry Pi, said in a blog post. "In even-numbered years, the conference is held at the MIT Media Lab, the birthplace of Scratch; in odd-numbered years, it takes place in other places around the globe.

"Since 2019 is also the launch year of Scratch 3, we think it's a fantastic opportunity for us to bring Scratch Conference Europe to the UK for the first time."

What you can look forward to:

- Hands-on, easy-to-follow workshops across a range of topics, including the new Scratch 3;
- Interactive projects to play with;
- Thought-provoking talks and keynotes;
- Plenty of informal chats, meetups, and opportunities for you to connect with other educators.

Join to become part of a growing community, discover how the Raspberry Pi Foundation can support you further, and develop your skills with Scratch as a creative tool for helping students learn to code.

Attend the conference:

Tickets for Scratch Conference Europe will go on sale in April.

For updates, subscribe to Raspberry Pi LEARN (**magpi.cc/THEdBx**), the monthly newsletter for educators, and keep an eye on **@Raspberry_Pi** on Twitter!

CONTRIBUTE TO SCRATCH CONFERENCE EUROPE

Would you like to contribute your own content at the event? The Foundation is looking for folks in the community to share or host:

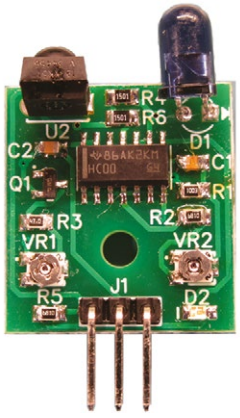
- Project demos
- Posters
- Workshops
- Discussion sessions
- Presentations
- Ignite talks

Young people under 18 will also be welcome as content contributors, but they must be supported by an adult. All content contributors will be able to attend the whole event for free. Find more details and apply to participate using this short online form: **magpi.cc/JtnZix**

▼ Scratch is a fantastic teaching tool

Crowdfund **this!**

Raspberry Pi projects you can crowdfund this month



IR Detector

This small and cheap add-on lets you add a tiny infrared sensor to your Pi projects. It uses open-source hardware and software and has some typical real-world uses such as a hand-drying sensor, security gates, proximity sensors, and more.

▶ **kck.st/2T8B3m0**

TickerTape

This retro alphanumeric display works with Raspberry Pi, Arduino, and other hardware. Like a lot of simple display add-ons, it has four buttons (which can be four touch points) and it's quite efficient as well. It has already hit its target as we go to print, so make sure to give it a look.

▶ **kck.st/2QisOBX**

CROWDFUNDING A PI PROJECT?

If you've launched an irresistible Pi-related project, let us know!

magpi@raspberrypi.org

Best of the rest!

Here are some other great things we saw this month

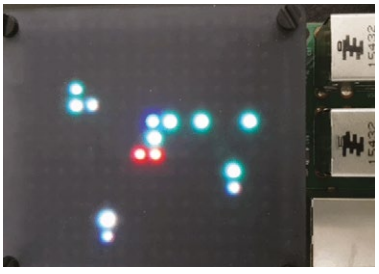
SUPERTUXKART ON PI!

While Tux, the penguin mascot for Linux, may not be as famous as Mario, it hasn't stopped people from using him in place of Nintendo's star character for an open-source clone of the Mario Kart series, called SuperTuxKart. It also now lets you create an online server on a Pi. Lovely.

▶ **magpi.cc/JxAUZe**

VISUAL AIRCRAFT TRACKER

Using a Unicorn HAT, this project lets you keep an eye on the skies at any location. This one is currently tracking over Heathrow, pulling info from online sources to create a lovely, and lively, activity. Must have been fun to watch during the drone incident.

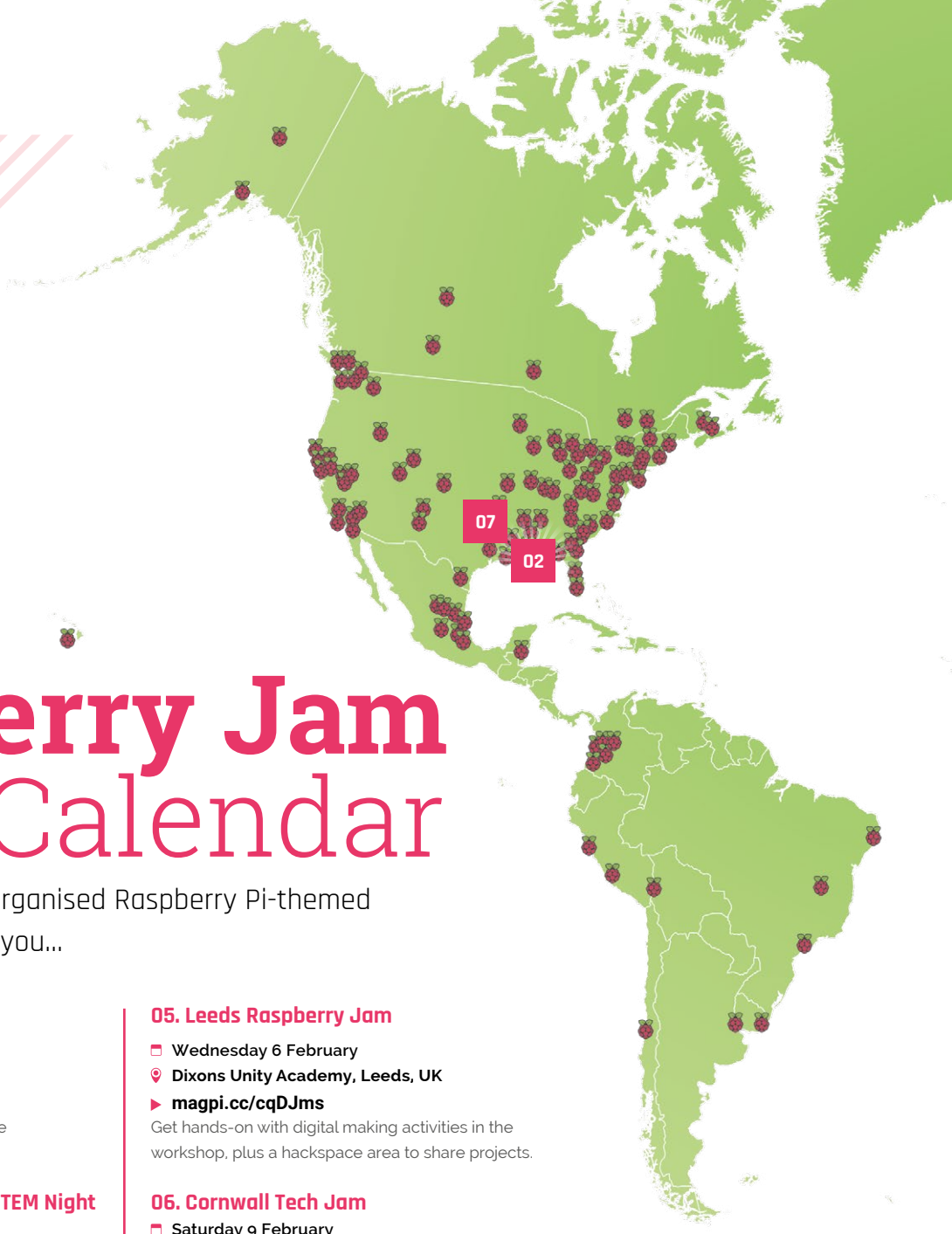


▶ **magpi.cc/pirevj**

GAME BOY ADVANCE SP

The GBA SP is one of the smallest handheld game systems that Nintendo ever made, making it a truly portable console. It's still big enough to fit a Raspberry Pi into the guts, though, allowing you to seriously power up your 15-year-old Game Boy.

▶ **magpi.cc/TzHEEf**



Raspberry Jam Event Calendar

Find out what community-organised Raspberry Pi-themed events are happening near you...

01. Raspberry Jam Kyiv

- Tuesday 12 February
- iLand, Kiev, Ukraine
- magpi.cc/XnisLa

Show off your projects or learn how to use your Raspberry Pi at this Ukrainian Jam!

02. Take Your Family to School STEM Night

- Tuesday 12 February
- Denton Magnet School, Mobile, AL, USA
- magpi.cc/mGMKDv

Dinner and Raspberry Pi for families of students at Denton Magnet School of Technology.

03. DCL Raspberry Pi Jam

- Wednesday 13 February
- Durham County Library, Durham, UK
- magpi.cc/UMbqih

Find out what the buzz is all about at DCL's first Raspberry Jam!

04. Raspberry Jam Big Birthday Weekend

- Saturday 2 March to Sunday 3 March
- Earth, Solar System, Milky Way
- magpi.cc/CrMUru

Get ready to celebrate the Raspberry Pi's birthday by going to your local Jam!

05. Leeds Raspberry Jam

- Wednesday 6 February
- Dixons Unity Academy, Leeds, UK
- magpi.cc/cqDJms

Get hands-on with digital making activities in the workshop, plus a hackspace area to share projects.

06. Cornwall Tech Jam

- Saturday 9 February
- Cornwall College Pool, Redruth, UK
- cornwalltechjam.uk

Learn about programming in various languages on a range of platforms including Arduino and Raspberry Pi.

07. Dallas Young Makers Club

- Saturday 23 February
- J. Erik Jonsson Central Library, Dallas, TX, USA
- dallasyoungmakers.org

Learn about computing and robotics on Raspberry Pi at this free workshop for kids!

08. Raspberry Pint

- Tuesday 26 February
- CodeNode, London, UK
- magpi.cc/iGrUiv

If you're working on a Pi project you'd like to share, or need some help with it, head down to Raspberry Pint!

FULL CALENDAR

Get a full list of upcoming events for February and beyond here:

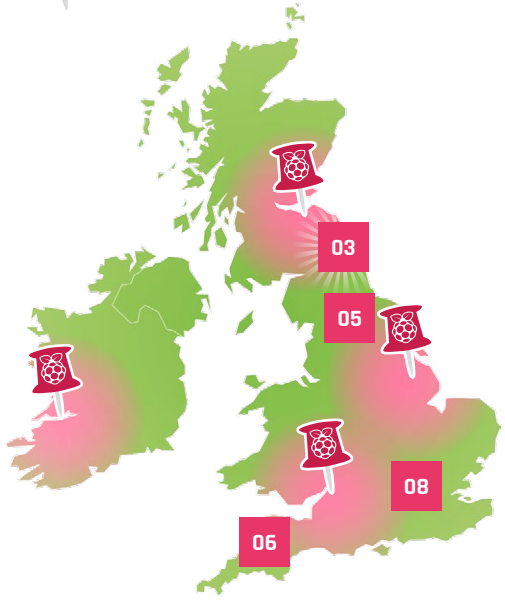
rpf.io/jam

FIND OUT ABOUT JAMS

Want a Raspberry Jam in your area? Want to start one? Email Ben Nuttall about it:

jam@raspberrypi.org

We've highlighted some of the areas in need of a Jam! Can you help out?



Raspberry Jam advice:

Starting a new Raspberry Jam

"Don't be afraid to start small: it helps you to gauge interest. Try somewhere out and find out what shape your Jam will eventually be."

Michael Horne – CamJam

Every Raspberry Jam is entitled to apply for a Jam starter kit, which includes magazine issues, printed worksheets, stickers, flyers, and more. Get the book here: **magpi.cc/2q9DHfQ**

Your Letters

▲ Raspberry Pi continues to sell very well; the Foundation just doesn't really talk about sales as much as it used to

3 for 5

I want to take advantage of the offer on your subscription page that says '…or try three issues for £5!'. I'm having some trouble trying to figure out how to do that – can you help me?

Josh via Facebook

The 3-for-5 offer can be found on our online store here: **magpi.cc/345**.

What you need to do is add the subscription to your cart, and add the code MP-SAVE at checkout. This will bring it down to £5, and you'll get the current issue, along with the following two issues, delivered straight to your door!

Contact us!

- Twitter **@TheMagPi**
- Facebook **magpi.cc/facebook**
- Email **magpi@raspberrypi.org**
- Online **raspberrypi.org/forums**

Magic numbers

Hello, I'm a French student. I'm actually making a project around the Raspberry Pi for my final exam for my degree, and I've contacted you because I can't find a lot of data about the sales of the Raspberry Pi in 2018 in the world, or in France.

Do people working in the Foundation have access to data or link to some data to give me so I can get a better idea of how well the Raspberry Pi is distributed in France and the rest of the world?

Antoine via email

Unfortunately, these numbers aren't really public, especially when it comes to specific regions. At this point we know the Raspberry Pi is definitely the third best-selling computer of all time (behind PCs and Macs), having sold tens of millions of units; however, the Raspberry Pi Foundation doesn't release regular figures about the worldwide sales.

If you send a nice email to the Raspberry Pi Foundation, though, the folks there might be able to tell you, but we can't promise anything!

Pi comparison

I am trying to get my kids excited about computers and coding. We are currently living in France, and unfortunately, unbelievably, and upsettingly, the French school system is terrible for computer learning. My kids go to private school and nothing – no classes!

So… we have a Raspberry Pi 3, two 2s, and a Zero… none have been used. I'm a parent that gets frozen, because I do not know where to start but I wanted to have the hardware to do so.

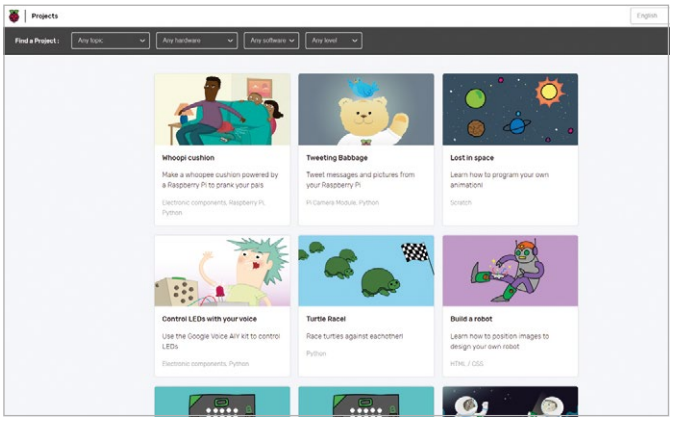
> "I'm a parent that gets frozen, because I do not know where to start but I wanted to have the hardware to do so"

At the same time, I notice that there are now newer versions of all our Pis. Is there any chart I can refer to to see the differences? Should I just throw away the old ones and buy new ones?

Bill via email

There's no official table of specs for the different Raspberry Pi models – although there are plenty of unofficial ones. The Wikipedia one is kept fairly up-to-date, and you can find it here: **magpi.cc/tXcpbP**.

However, don't throw your old Pi computers away! All the official Raspberry Pi resources and projects, and all the tutorials here in *The MagPi*, will work on just about any Raspberry Pi. You can definitely learn to code (whether you're a kid or adult) on one of the Raspberry Pi models you have listed. We'd recommend to do it with the Pi 3 if you can.



▼ The Raspberry Pi projects website is filled with guides and resources that work on just about every single Raspberry Pi

▲ There are Raspberry Jams and other events going on around the world

Project helpers

I would like help with Raspberry Pi projects, but I'm worried about doing it wrong and maybe damaging something. I would love to know if someone who uses Raspberry Pis and can help me with this lives near me. Help!

Maclean via email

There's no real database of specific Pi people we can point you towards, but don't despair! We have a few tips on how to find people in your area.

First of all, there is the official map of Raspberry Jams which you can find here: **rpf.io/jam**.

From there you can see if there are any events, such as Jams, going on in your area. Usually, these Jams are also connected to a local community of makers and Pi tinkerers that you can probably join up with.

We also suggest to try looking on Facebook for a Raspberry Pi community group in your area (or country). A lot of countries and regions have one, and from there you can find people more local to you who might be willing to help!

Finally, check out the Raspberry Pi Forums and see if there's anyone local to you that wants to help, or might just be willing to help online!

We hope you find someone.

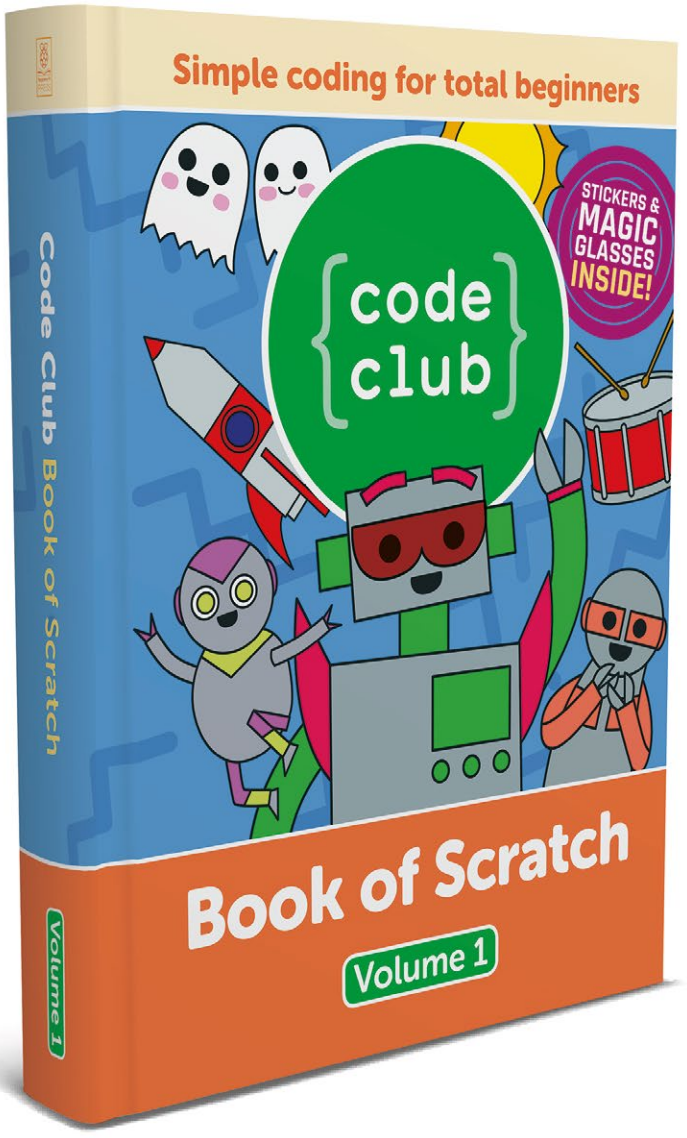
£9.99
Free worldwide shipping!

code club

Book of Scratch

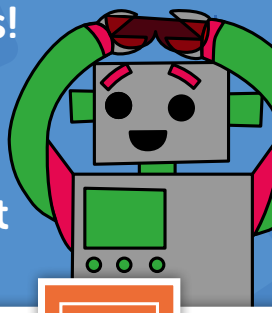
Volume 1

Simple coding for total beginners



The first Code Club book has arrived!

- Learn to code using Scratch, the block-based language
- Follow step-by-step guides to create games and animations
- Use the magic glasses to reveal secret hints
- Includes 24 exclusive Code Club stickers!
- The special spiral binding allows the book to lay flat



Available at: magpi.cc/CCbook1

Raspberry Pi PRESS

WIN One of Two PiDP-11 retro computer kits

> The PiDP-11 is a modern replica of the PDP-11/70

Obsolescence Guaranteed

We reviewed the PiDP-11 last issue, loving its straightforward build and beautifully accurate case. Get your hands on one with this competition. We have two kits to give away.

Head here to enter: magpi.cc/win | **Learn more:** magpi.cc/ZTUNAL

Terms & Conditions

Competition opens on **30 January 2019** and closes on **28 February 2019**. Prize is offered to participants worldwide aged 13 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from The MagPi magazine. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram or Facebook.

HackSpace

TECHNOLOGY IN YOUR HANDS

THE NEW MAGAZINE FOR THE MODERN MAKER

ISSUE #15

OUT NOW

hsmag.cc

Available on the App Store

GET IT ON Google Play

NEXT MONTH

AFFORDABLE PROJECTS

Learn for less with our modestly priced makes

#79 ON SALE 28 FEB

PLUS!

- DEVELOP AN ANDROID APP
- BUILD A KARAOKE MACHINE
- DETECT LICENCE PLATE NUMBERS
- TOP 10 HOME AUTOMATION PROJECTS

DON'T MISS OUT!

magpi.cc/subscribe

TWITTER	@TheMagPi
FACEBOOK	fb.com/MagPiMagazine
EMAIL	magpi@raspberrypi.org

EDITORIAL

Editor
Lucy Hattersley
lucy@raspberrypi.org

Features Editor
Rob Zwetsloot
rob.zwetsloot@raspberrypi.org

Sub Editors
Phil King and Nicola King

DESIGN

criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Mike Kay, Harriet Knight

Illustrator
Sam Alder

CONTRIBUTORS

Mike Cook, David Crookes, PJ Evans, Rosie Hattersley, Nicola King, Ben Nuttall, Ross O'Neill, KG Orphanides, Marc Scott, Mark Vanstone

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.org
+44 (0)7904 766523

Director of Communications
Liz Upton

CEO
Eben Upton

DISTRIBUTION

Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS

Raspberry Pi Press
Mann Enterprises, Unit E,
Brocks Business Centre,
Haverhill, CB9 8QP

To subscribe
magpi.cc/subscribe

To get help:
rpipresshelp@raspberrypi.org

recycle

This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

The MagPi magazine is published by Raspberry Pi (Trading) Ltd., Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products, or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0).

Raspberry Pi PRESS

ISSN: 2051-9982.

Bring coding to your community

Programme Manager **Ross O'Neill** on growing CoderDojo in the UK

Clicking, creative chattering, and coding: typical sounds now heard every Saturday morning in many of our previously hushed library spaces across the UK. A hustle and bustle of friend-forging, project learning, and sometimes keyboard-bashing activity – that's what CoderDojo is for me.

I first heard about CoderDojo in 2014 within the halls of a student residence at Trinity College Dublin, Ireland. "It's called 'CoderDojo'," I was told by a friend of a friend at the time, Rosa Langhammer – now the General Manager of the CoderDojo Foundation. But back then she was an enthusiastic tiger-striped onesie-wearing graduate onboarding a prospective new volunteer.

"CoderDojo?" I replied – a name I soon found out represented an open-source global community of *free* volunteer-led clubs across the world for young people. "I wish I'd had that when I was kid!" I thought. "Sign me up!" I cried. Rosa guided me to 'The Map': a colourful, vibrant, pin-filled map, each pin representing a regular running club in a vast and diverse set of venues, from community halls and libraries to large corporate offices across the world. I became a CoderDojo 'Champion', the organiser of what was (in my mind) the smallest Dojo in the world – just four young kids in a local community centre in The Liberties, Dublin. I later found that wasn't unusual. We grew quickly to a club of 25 young people, with the help of other volunteers. I specifically say 'we grew' because it was a team effort, of complete strangers brought together with a passion for technology.

> I want to make it as easy as possible for any energetic prospective volunteer who raises their hand to get started

Increasing opportunities

That was a long time ago and today I work for the CoderDojo Foundation, part of the Raspberry Pi Foundation family. One of my main aims is to get Dojos opened in as many library spaces as possible across the UK. I want to make it as easy as possible for any energetic prospective volunteer who raises their hand to get started. I'm working directly with local authorities, library volunteer coordinators, and those individual volunteers on a one-to-one basis – a strategy my inner algorithmic nerd is calling 'scalable brute force'.

The team and I also like to work with other organisations with similar missions, values, and passions. This year we're working directly with different hubs and groups to help onboard new volunteers to implement STEM-related activities such as CoderDojo; doing presentations, training, and onboarding sessions across the UK. I feel like our volunteer opportunities need to be showcased alongside some of the best and most reputable organisations online. My team's hope is to increase the number of opportunities on a variety of volunteer platforms tenfold over the next number of months.

That's me, that's my work and my passion, but I am more interested in you, the reader. Like me, do you also wish you had CoderDojo when you were a kid? Why not start your own? Let's talk! Volunteer at: **coderdojo.com/volunteer-uk**.

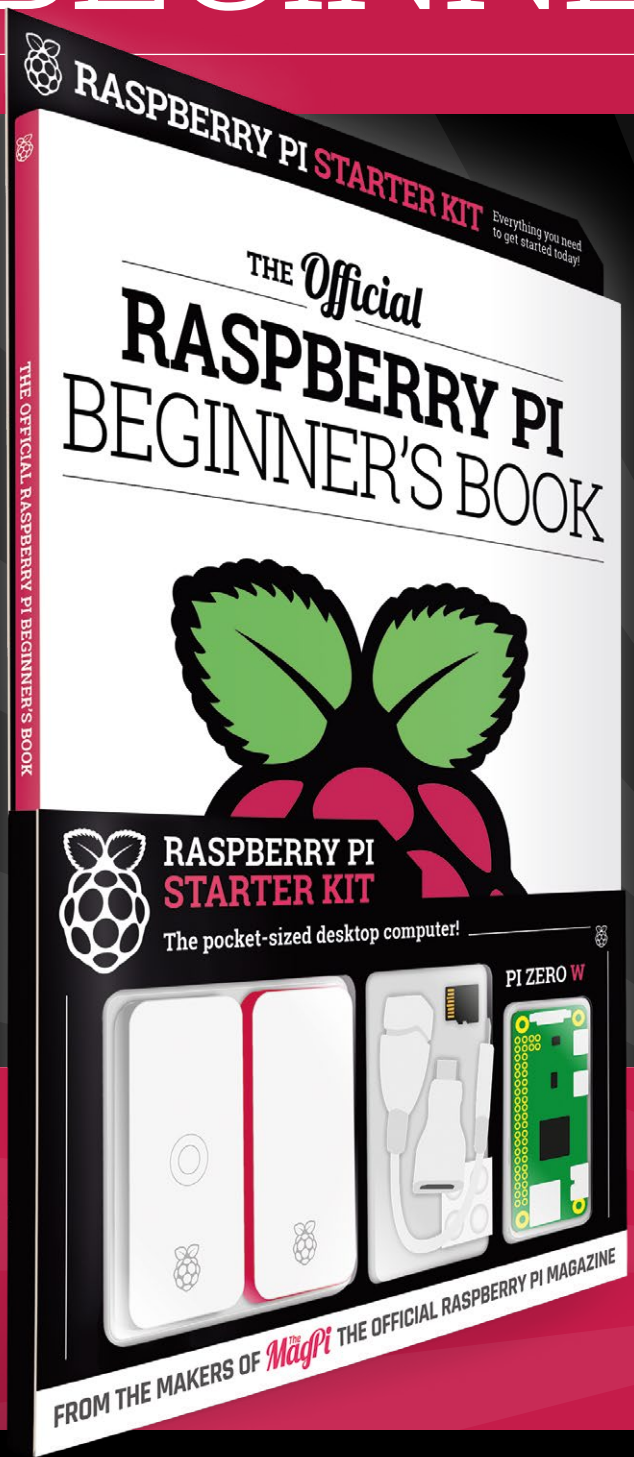
AUTHOR

Ross O'Neill

As well as being the Programme Manager for CoderDojo, Ross is an occasional journalist, coder, and storyteller.

ross@coderdojo.org
coderdojo.com

THE *Official* RASPBERRY PI BEGINNER'S BOOK


RASPBERRY PI STARTER KIT
Everything you need to get started today!
THE Official RASPBERRY PI BEGINNER'S BOOK
THE OFFICIAL RASPBERRY PI BEGINNER'S BOOK
RASPBERRY PI STARTER KIT
The pocket-sized desktop computer!
PI ZERO W
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE

LEARN COMPUTING THE EASY WAY!

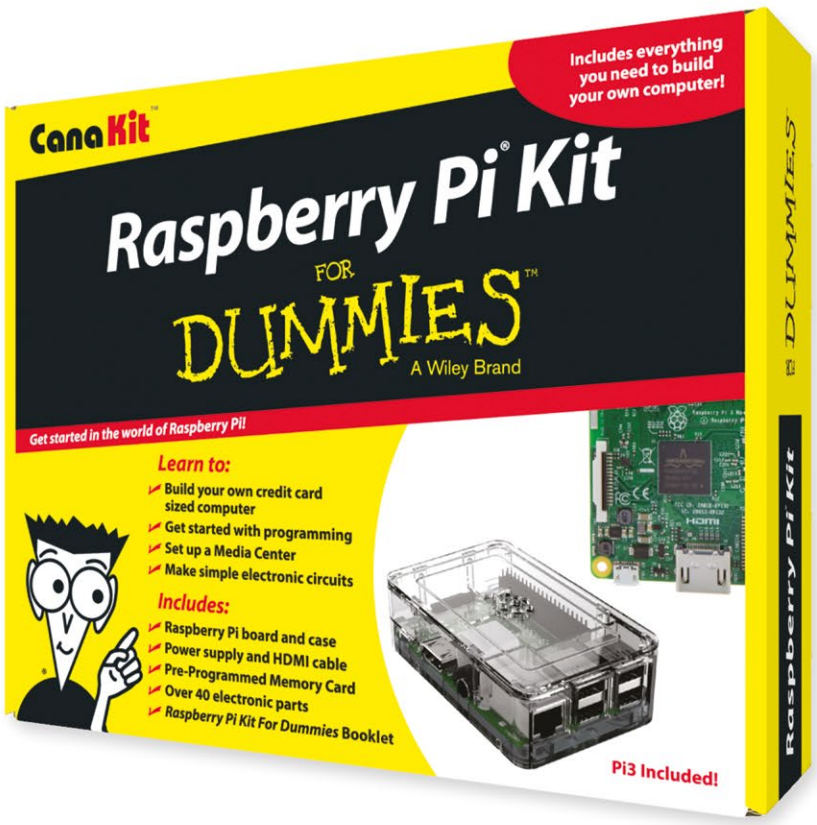
Includes

- Pi Zero W computer
- Official case with three covers
- USB and HDMI adapters
- 8GB microSD card
- 116-page beginner's book

Available now

Buy online: magpi.cc/store

CanaKit™


CanaKit
Includes everything you need to build your own computer!
Raspberry Pi® Kit
FOR DUMMIES™
A Wiley Brand
Get started in the world of Raspberry Pi!
Learn to:
Build your own credit card sized computer
Get started with programming
Set up a Media Center
Make simple electronic circuits
Includes:
Raspberry Pi board and case
Power supply and HDMI cable
Pre-Programmed Memory Card
Over 40 electronic parts
Raspberry Pi Kit For Dummies Booklet
Pi3 Included!
Raspberry Pi Kit
FOR DUMMIES

FOR DUMMIES® A Wiley Brand

Available for worldwide shipping at:

WWW.CANAKIT.COM

Available in Europe through RS Components

RS

Kit Includes:

- *Raspberry Pi For Dummies* Booklet
- Raspberry Pi 3 Board
- Memory Card
- Plastic Case
- 2.5A Power Supply
- HDMI Cable
- Resistors
- LEDs
- Push Button Switches
- Prototyping Breadboard
- Jumper Wires
- Heat Sinks

$89.99 US DOLLARS

£69.99 EXCLUDING VAT

Raspberry Pi is a registered trademark of the Raspberry Pi Foundation. For Dummies and the Dummies Man logo are trademarks or registered trademarks of John Wiley & Sons, Inc. Used under license. RS logo is a registered trademark of RS Components Ltd. CanaKit is a registered trademark of Cana Kit Corporation.

ELECTRONIC KITS • ELECTRONIC PARTS • RASPBERRY PI • ARDUINO